

UNIVERZA V LJUBLJANI  
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Simon Hočevar

**Poučevanje računalništva z LEGO MINDSTORMS**

DIPLOMSKO DELO

VISOKOŠOLSKI STROKOVNI ŠTUDIJSKI PROGRAM PRVE STOPNJE  
RAČUNALNIŠTVO IN INFORMATIKA

Ljubljana, 2017



UNIVERZA V LJUBLJANI  
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Simon Hočevar

**Poučevanje računalništva z LEGO MINDSTORMS**

DIPLOMSKO DELO

VISOKOŠOLSKI STROKOVNI ŠTUDIJSKI PROGRAM PRVE STOPNJE  
RAČUNALNIŠTVO IN INFORMATIKA

MENTOR: izr. prof. dr. Janez Demšar

Ljubljana, 2017



To delo je ponujeno pod licenco *Creative Commons Priznanje avtorstva – Deljenje pod enakimi pogoji 2.5 Slovenija* (ali novejšo različico). To pomeni, da se tako besedilo, slike, grafi in druge sestavine dela kot tudi rezultati diplomskega dela lahko prosto distribuirajo, reproducirajo, uporabljajo, priobčujejo javnosti in predelujejo pod pogojem, da se jasno in vidno navede avtorja in naslov tega dela in da se v primeru spremembe, preoblikovanja ali uporabe tega dela v svojem delu lahko distribuira predelava le pod licenco, ki je enaka tej. Podrobnosti licence so dostopne na spletni strani [creativecommons.si](http://creativecommons.si) ali na Inštitutu za intelektualno lastnino, Streliška 1, 1000 Ljubljana.



Izvorna koda diplomskega dela, njeni rezultati in v ta namen razvita programska oprema je ponujena pod licenco *GNU General Public License*, različica 3 (ali novejša). To pomeni, da se lahko prosto distribuira in/ali predeluje pod njenimi pogoji. Podrobnosti licence so dostopne na spletni strani <http://www.gnu.org/licenses>.



Fakulteta za računalništvo in informatiko izdaja naslednjo nalogo:

*Poučevanje računalništva z Lego Mindstorms (ang. Teaching Computer Science with Lego Mindstorms)*

Tematika naloge:

*V okviru predmeta gimnazijskega predmeta Informatika želimo dijake naučiti tudi osnov programiranja, kot enega od načinov za spoznavanje splošnejših konceptov računalniškega razmišljanja. Pri tem naletimo na problem nizke motivacije, saj je programiranje lahko abstraktno in dolgočasno.*

*Ena od možnih rešitev je "fizično računalništvo". V diplomski nalogi sestavite možen potek poučevanja programiranja prek manjših projektov z roboti Lego Mindstorms.*

*"Informatics" is a high school subject that should include basics of programming, with a goal of introducing highschoolers to some general principles of algorithmic thinking. The problem that we usually encounter is the lack of motivation since programming can be abstract, dry and boring.*

*A possible solution is to teach "physical computing". Propose a set of activities for teaching programming by using Lego Mindstorms robots.*





## IZJAVA O AVTORSTVU DIPLOMSKEGA DELA

Spodaj podpisani Simon Hočevar sem avtor diplomskega dela z naslovom:

*Poučevanje računalništva z LEGO MINDSTORMS (ang. Teaching Computer Science with LEGO MINDSTORMS)*

S svojim podpisom zagotavljam, da:

- sem diplomsko delo izdelal samostojno pod mentorstvom izr. prof. dr. Janeza Demšarja,
- so elektronska oblika diplomskega dela, Poučevanje računalništva z LEGO MINDSTORMS povzetek (ang. Teaching computer science with LEGO MINDSTORMS) ter ključne besede gimnazija, računalništvo, informatika, kurikulum in LEGO MINDSTORMS (ang. secondary school, computer science, informatics, curriculum, LEGO MINDSTORMS) identični s tiskano obliko diplomskega dela,
- soglašam z javno objavo elektronske oblike diplomskega dela na svetovnem spletu prek univerzitetnega spletnega arhiva.

V Ljubljani, dne 13. 2. 2017

Podpis avtorja:







*Zahvaljujem se svojemu mentorju izr. prof. dr. Janezu Demšarju za pomoč pri  
izdelavi te diplomske naloge.*

*Zahvaljujem se tudi Gimnaziji Novo mesto in vsem zaposlenim, ki so mi omogočili  
izvajanje predmeta Informatika in mi pomagali do boljših rezultatov ter boljših  
učnih priprav, ki so glavni del te naloge.*

*Zahvaljujem se svoji družini, ki mi je ves čas stala ob strani, me podpirala in  
bodrila.*



# Kazalo

**Povzetek**

**Abstract**

<b>Poglavje 1</b>	<b>Uvod .....</b>	<b>1</b>
<b>Poglavje 2</b>	<b>Informatika v gimnaziji .....</b>	<b>4</b>
<b>Poglavje 3</b>	<b>LEGO MINDSTORMS.....</b>	<b>7</b>
3.1	Razvoj LEGO MINDSTORMS.....	7
3.2	Kompleti EV3 .....	8
3.3	Kocka EV3.....	11
3.4	Motorji .....	12
3.5	Senzorji .....	12
3.5.1	Barvni senzor.....	13
3.5.2	Žiro senzor .....	14
3.5.3	Pritisni senzor .....	15
3.5.4	Ultrazvočni senzor.....	15
3.5.5	Infrardeči senzor in infrardeči daljinski upravljalnik .....	15
3.5.6	Infrardeči daljinski upravljalnik .....	16
3.5.7	Temperaturni senzor .....	16
3.6	Programska oprema EV3 .....	16
3.6.1	Pregled osnovnih funkcij programske opreme EV3.....	17
3.6.2	Programiranje z bloki .....	20
<b>Poglavje 4</b>	<b>Učne priprave izbirnega predmeta Informatika .....</b>	<b>26</b>
4.1	Potrebščine za izvajanje predmeta .....	27
4.2	Uvod v programiranje .....	28
4.3	Programiranje robotov LEGO Mindstorms .....	29

4.4	Ideje za bogatenje pouka .....	32
<b>Poglavje 5</b>	<b>Zaključek .....</b>	<b>33</b>



## Seznam uporabljenih kratic

kratica	angleško	slovensko
CS	Computer Science	računalništvo
LM	LEGO MINDSTORMS	LEGO MINDSTORMS



## **Povzetek**

**Naslov:** Poučevanje računalništva z LEGO MINDSTORMS

V diplomskem delu je predstavljen način poučevanja izbirnega predmeta Informatika v programu gimnazije. V okviru diplomskega dela so pripravljene učne priprave za uporabo robotov LEGO MINDSTORMS pri pouku.

Oblikovane vsebine se osredotočajo predvsem na programiranje, hkrati pa se navezujejo še na druga tematska področja predmeta Informatika v gimnazijah. Učne priprave so pripravljene v skladu s smernicami veljavnega učnega načrta za predmet Informatika v gimnazijah in s smernicami mednarodnega kurikulumu za poučevanje računalništva (Computer Science Curricula 2013). V pripravih so zavzete tudi teme in primeri, ki so primerni za informatiko na maturi.

**Ključne besede:** gimnazija, računalništvo, informatika, kurikulum, LEGO MINDSTORMS



## **Abstract**

**Title:** Teaching Computer Science with LEGO MINDSTORMS

This document presents a way of teaching the elective course of informatics in a secondary school. Lesson plans for LEGO MINDSTORMS robot use at classes forms part of this paper as well.

The content focuses mainly on programming, however it also refers to other themes that constitute a part of Informatics class in a secondary school. Lesson plans were prepared in accordance with the guidelines of the currently valid curriculum for Informatics class in secondary schools and the guidelines of the international curriculum for teaching computer science (Computer Science Curricula 2013). They also include themes and examples that are considered useful for the students' final exam *matura*.

**Keywords:** secondary school, computer science, informatics, curriculum, LEGO MINDSTORMS



## Poglavje 1      Uvod

V današnji družbi ima računalništvo pomembno vlogo, saj je vključeno v vsa področja našega življenja. Da bi bili v tej družbi uspešni, moramo razumeti delovanje informacijsko-komunikacijskih tehnologij in konceptov, na katerih temelji tehnologija. [1]

Diplomsko delo je nastajalo ob poučevanju predmeta Informatika na gimnaziji. Med poučevanjem sem ugotovil, da dijaki v prvem letniku gimnazije dobro rokujejo z računalnikom. Znano se povezati na svetovni splet, klikati in tipkati. Večina pa jih ne ve, kako računalnik deluje in kakšni so osnovni principi. Sicer vedo, da če kliknejo na ustrezen gumb, se bodo podatki, ki so zbrani na ekranu, razvrstili v abecednem redu, ne vedo pa, kaj se ob njihovem kliku zgodi v ozadju.

Učenje digitalne pismenosti se v današnji družbi začne že pri otroku. Današnji otroci so namreč pogosto izpostavljeni računalnikom, velikokrat pa še kakšnim drugim elektronskim napravam, naj bo to mobilni telefon, tablični ali pa namizni računalnik. Pri današnjih otrocih opažamo, da zelo hitro dojemajo osnovne principe delovanja sodobnih elektronskih naprav. Če imajo starejši ljudje težave z upravljanjem računalniške miške in zaslonov na dotik, pa pri otrocih opažamo, da vse te stvari dojemajo izjemno hitro in brez večjih težav.

Digitalno opismenjevanje se po otroškem raziskovanju raznovrstnih naprav nadaljuje v vrtcu. V mnogih slovenskih vrtcih otroci za igro poleg klasičnih igrač že uporabljajo tudi tablične in namizne računalnike. Večinoma sicer zgolj za igro in razvedrilo, vendar se navsezadnje otrok skozi igro tudi veliko nauči. V tej fazi otrok osvoji predvsem različne načine uporabe naprav in varnega rokovanja z njimi.

V osnovni šoli začnejo otroci uporabljati digitalne naprave tudi kot učne pripomočke. Povežejo se v svetovni splet, komunicirajo z drugimi uporabniki, iščejo informacije itd. Nekatere osnovne šole v 4., 5. in 6. razredu ponujajo tudi neobvezni izbirni predmet Računalništvo, ki učence seznaja z različnimi področji računalništva. Predmet ne temelji na spoznavanju dela s posameznimi programi, ampak učence seznaja s temeljnimi računalniškimi koncepti in procesi. Učenci se pri pouku seznajajo s tehnikami in metodami reševanja problemov ter razvijajo algoritmičen način razmišljanja, spoznavajo omejitve računalnikov in njihov vpliv na družbo. Takšen način dela spodbuja ustvarjalnost, sodelovanje in poseben način razmišljanja

ter delovanja. S spoznavanjem računalniških konceptov in razvijanjem postopkovnega načina razmišljanja učenci pridobivajo znanja, spretnosti in veščine, ki so znatno trajnejši kot hitro razvijajoče se tehnologije. Ta znanja so neodvisna od tehnologij. Znanja, ki jih pridobijo učenci pri tem predmetu, so prenosljiva in uporabna na vseh področjih človeških dejavnosti. Učencem bodo koristila tudi pri drugih predmetih in v poznejšem življenju. Ta znanja, spretnosti in veščine so del digitalne pismenosti v evropskem referenčnem okvirju osmih ključnih kompetenc. [1]

Diplomsko delo se osredotoča predvsem na gimnazijske programe. Dijak, ki je vpisan v gimnazijski program, naj bi imel določena predznanja in naj bi že imel razvitih nekaj lastnosti digitalne pismenosti, ki jih je pridobil skozi dosedanje izobraževanje. V prvem letniku gimnazijskega programa dijaki obiskujejo obvezni predmet Informatika.

Informatika je splošnoizobraževalni predmet, pri katerem se teorija poznavanja in razumevanja osnovnih zakonitosti informatike prepleta z metodami neposrednega iskanja, zbiranja, hranjenja, vrednotenja, obdelave in uporabe podatkov z digitalno tehnologijo z namenom oblikovanja relevantnih informacij za dograjevanje lastnega znanja in za njegovo predstavitev oziroma posredovanje drugim. Predmet dijakom odpira možnost, da sistematično razvijajo digitalno kompetenco, ki je nujna za razumevanje digitalne tehnologije in procesov ter kakovostno uporabo te tehnologije z namenom omogočiti dijakom uspešen študij na univerzi in orientacijo za nadaljnje življenje. [2]

Ves čas izobraževanja sta v ospredju aktivna vloga dijakov in njihov osebni, strokovni ter jezikovni razvoj. Skupinsko delo pri projektih, problemsko in sodelovalno učenje, izbiranje vsebin glede na zanimanje in sposobnost dijakov ter upoštevanje njihovih idej, individualizacija, vključevanje raznih socialnih dejavnosti, uporaba debatnih metod, povezovanje med predmeti, timsko poučevanje z učitelji drugih predmetov in šolskim knjižničarjem ter zunanjimi strokovnjaki, sodelovanje z drugimi šolami in ustanovami doma in v tujini ter razvijanje strategij mišljenja so temeljne oblike dela pri predmetu. Pri tem se razvija sposobnost ustvarjalnega in kritičnega mišljenja ter presojanja, da bi zagotovili razumno in samozavestno odločanje dijakov v novih in nepredvidljivih okoliščinah. [2]

Del predmeta Informatika v gimnaziji je tudi učenje programiranja. Dijaki se učijo osnov programiranja v različnih okoljih z različnimi orodji. Pri poučevanju programiranja naletimo predvsem na težavi pomanjkanja zanimanja dijakov in pomanjkanja fizične predstave o funkcijah, ukazih, vejitvah, zankah ipd. Izkaže se, da se dijaki hitreje učijo in so dovzetnejši, če imajo pravo predstavo o tem, kaj so ustvarili, in ne zgolj fiktivno na ekranu ali samo s številkami. Zato smo pri predmetu Informatika uporabili robote LEGO MINDSTORMS EV3.



V diplomskem delu je predstavljen način poučevanja izbirnega predmeta Informatika v gimnaziji. Da bi dijaki bolje razumeli osnovno delovanje računalnika in računalniških programov, sem se odločil za poučevanje s pomočjo robotov LEGO MINDSTORMS EV3. V nadaljevanju so najprej opisane glavne karakteristike omenjenih robotov, senzorjev in motorjev ter programske opreme za programiranje.

Osrednji del diplomskega dela so izdelane učne priprave za izbirni predmet Informatika v drugem letniku gimnazijskega programa z natančnimi učnimi pripravami za nekaj posameznih učnih ur. Roboti LEGO MINDSTORMS EV3 so v učnih pripravah uporabljeni kot osrednje orodje za učenje programiranja in reševanje problemov. Priprave so izdelane v skladu z veljavnim učnim načrtom za predmet Informatika v gimnaziji in kurikulumom »Computer Science Curricula 2013« in so delno primerne za uporabo tudi pri neobveznem izbirnem predmetu Računalništvo v osnovni šoli.

Učne priprave so nastale ob poučevanju predmeta Informatika na Gimnaziji Novo mesto. Na šoli sem poučeval tri leta, od tega dve leti tudi izbirni predmet. V tem času sem učne priprave tudi izpopolnjeval in tako predstavljajo plod sprotnih priprav ter so že prilagojene v skladu s pripombami dijakov in sprotnimi opažanji.

## Poglavje 2 Informatika v gimnaziji

Predmet Informatika je v programu splošne, klasične ali strokovne gimnazije razdeljen na dva dela:

- obvezni predmet, ki se praviloma izvaja v prvem letniku in zavzema 70 pedagoških ur in
- izbirni predmet, ki se izvaja v drugem, tretjem ali četrtem letniku in zavzema 210 pedagoških ur.

Dijak lahko opravlja izbirni predmet Informatika na maturi, če ima opravljen tako obvezni kot izbirni predmet, torej skupno 380 pedagoških ur predmeta Informatika.

**Izbirni predmet**, ki zavzema 210 pedagoških ur, se praviloma razdeli v več letnikov. Večina slovenskih gimnazij razdeljuje izbirni predmet v drugi in četrty letnik. Tako dijaki v drugem letniku opravijo 70 ur izbirnega predmeta (2 uri tedensko), v četrtem letniku pa kot izbirni maturitetni predmet opravijo še 140 ur (4 ure tedensko). Predmet je priporočeno izvajati v dvojnih t. i. blok urah, saj se s tem doseže boljša učinkovitost predmeta.

Osrednji del tega diplomskega dela (učne priprave) se nanašajo na del izbirnega predmeta Informatika, ki se izvaja v drugem letniku in zavzema 70 pedagoških ur. [2]

**Splošni cilji in kompetence** predmeta Informatika so naravnani tako, da dijaki sistematično razvijajo znanje, veščine, spretnosti, osebnostne in vedenjske značilnosti, prepričanja, motive ter druge zmožnosti digitalne kompetence. [2]

Pri tem širijo in osmišljajo več kompetenc:

- kompetenco sporazumevanja v maternem jeziku,
- kompetenco sporazumevanja v tujih jezikih,
- matematično kompetenco in osnovno kompetenco v znanosti in tehnologiji,
- kompetenco učenja učenja,

- socialno in državljansko kompetenco,
- kompetenco samoiniciativnosti in podjetnosti,
- kompetenco kulturne zavesti in izražanja.

Kompetence in cilje predmeta dijaki dosejajo na različne načine, npr. z iskanjem in vrednotenjem podatkov ter informacij v maternem in tujem jeziku, reševanjem avtentičnih informacijskih problemov ter izdelavo in predstavitev rešitev informacijskega problema.

**Cilji in vsebine** predmeta so razdeljeni v dve skupini: splošna znanja in posebna znanja. Vsaka skupina je opredeljena s tematskimi sklopi, v okviru katerih se obdelajo posamezne vsebine predmeta.

**Splošna znanja** so razdeljena v tri tematske sklope: osnove informatike, digitalna tehnologija in predstavitev informacij.

Glede osnov informatike se dijaki predvsem naučijo in opredelijo temeljne pojme informatike, kot so podatek, informacija, znanje, računalništvo, informatika itd. Spoznajo tudi družbene vidike informatike in opredelijo komuniciranje, razložijo njegov pomen ter cilje.

V tematskem sklopu digitalne tehnologije dijaki spoznajo namen, vlogo in pomen digitalne tehnologije, opredelijo, kdaj in kje lahko uporabijo določeno tehnologijo, spoznajo mejnike razvoja računalništva, zgradbo ter delovanje računalnika, strojno in programsko opremo ter razložijo pojme računalniških omrežij in opredelijo njihovo delovanje.

V sklopu predstavitve informacij dijaki spoznajo osnove in pomen zapisa podatkov, različne standarde in načine predstavitve informacij ter pridobivanje podatkov iz različnih virov, vrednotijo vire itd.

Posebna znanja so razdeljena v dva sklopa: predstavitev informacij in obdelava podatkov.

V sklopu predstavitve informacij se dijaki naučijo predstavljati informacije na različne načine – pisno, slikovno, zvočno, z gibljivo sliko, računalniškimi prosojnicami in informacijami na svetovnem spletu. Pri vsakem načinu opredelijo pomen in prednosti ter slabosti tega načina predstavitve ter se naučijo njihovih osnovnih lastnosti.

V sklopu obdelave podatkov dijaki opredelijo pojme računalniške obdelave podatkov, algoritma, programskih jezikov, programiranja, podatkovne baze, preglednic in tehnologij znanja.

Izbirni predmet Informatika v drugem letniku se izvaja dve uri tedensko; praviloma je to ena dvojna oz. blok ura tedensko. Skupno izbirni predmet v enem šolskem letu zavzema 70 pedagoških ur.

Pri tem predmetu ne gre za neposredno nadaljevanje izvajanja iz obveznega predmeta Informatika v prvem letniku, a se kljub temu tesno navezuje na omenjeni predmet. Cilji in kompetence obveznega in izbirnega predmeta so identični, le v drugačnih obsežnostih in z drugačnimi orodji. Predmet se navezuje predvsem na posebna znanja iz učnega načrta predmeta Informatika v gimnazijah, saj se splošna znanja praviloma že obdelajo v okviru obveznega predmeta v prvem letniku.

## **Poglavje 3     LEGO MINDSTORMS**

LEGO MINDSTORMS (v nadaljevanju: LM) je serija posebnih programskih okolij in posebne strojne opreme podjetja LEGO, s katerimi gradimo prilagodljive robote. Kompleti LM vsebujejo inteligentno kocko (ang. intelligent brick), senzorje, motorje in LEGO kocke iz LEGO linije Technic. Z njihovo pomočjo gradimo zapletene pametne mehanične sisteme. [3]

### **3.1     Razvoj LEGO MINDSTORMS**

Razvoj LM sega v leto 1986, ko je podjetje LEGO izdalo svoje prve produkte LEGO Technic, ki so bili krmiljeni s pomočjo računalnika.

Leta 1988 je podjetje LEGO v sodelovanju z inštitutom MIT (Massachusetts Institute of Technology) pričelo z razvojem t. i. inteligentne kocke (ang. intelligent brick), ki bi bila programabilna.

Deset let pozneje sta bila v Muzeju moderne umetnosti v Londonu predstavljena inteligentna kocka »LEGO MINDSTORMS RCX Intelligent Brick« in sistem »Robotics Invention System«. To je bil prvi sistem oz. komplet z inteligentno kocko, senzorji in motorji, kot jih poznamo danes v kompletih Mindstorms. Septembra 1998 je bil sistem tudi ponujen v prodajo, in sicer hkrati v Združenih državah Amerike in Združenem kraljestvu. Novembra sta se lastnik skupine LEGO, Kjeld Kirk Kristiansen, in ustanovitelj organizacije FIRST (For Inspiration and Recognition of Science and Technology), Dean Kamen, povezala in ustanovila FLL (FIRST LEGO League), tekmovanje v programiranju LEGO robotov za učence in dijake. Prvo tekmovanje se je odvijalo v Chicagu, na njem pa je sodelovalo 200 ekip. Tekmovanje vsakoletno poteka še danes. Razširilo se je na svetovno raven in v njem sodelujejo tudi slovenske ekipe učencev in dijakov.

Leta 2006 so komplet Robotics Invention System popolnoma prenovili, dopolnili in izpopolnili ter mu nadeli novo ime: LEGO MINDSTORMS NXT. Tri leta pozneje je NXT dobil svojega naslednika – NXT 2.0.

Leta 2013, ob 15. obletnici razvoja LM, so izdelali popolnoma nov LM komplet, poimenovan LEGO MINDSTORMS EV3. [4] To je torej tretja generacija kompletov LM. Če razčlenimo ime EV3, dobimo kratico EV – evolucija (ang. evolution) in številko 3 – tretja generacija. [5]

Slika 1 prikazuje primerjavo med prvo, drugo in tretjo inteligentno kocko LEGO Mindstorms – RCX iz leta 1998, NXT iz leta 2006 in EV3 iz leta 2013.

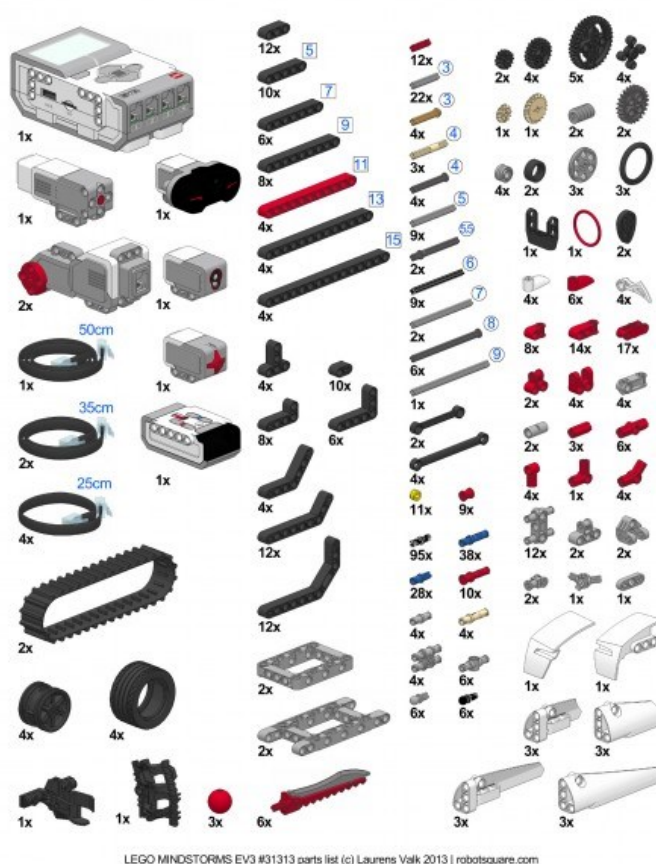


Slika 1: Primerjava RCX, NXT in EV3 inteligentnih kock [6]

## 3.2 Kompleti EV3

Obstajata dva različna osnovna kompleta LM EV3: Home Edition in Education Core Set. Kompleta se razlikujeta po namembnosti in vsebini. Poleg osnovnih kompletov so na voljo še razširitveni in tematski kompleti.

**Home Edition** je namenjen domačim uporabnikom in je bolj »komercialna« različica kompleta EV3. Kocko programiramo z brezplačnim programskim okoljem, robota pa lahko vodimo s pomočjo pametnega telefona ali priloženim infrardečim daljinskim upravljalnikom. Slika 2 prikazuje vsebino kompleta: EV3 kocko, dva velika motorja, srednji motor, pritisni senzor, barvni senzor, infrardeči senzor, infrardeči daljinski upravljalnik, 600 gradnikov, podlogo za igranje in načrte za grajenje petih različnih robotov. Komplet je na trgu prosto dostopen. [7]



Slika 2: Vsebina kompleta EV3 Home Edition [7]

**Education EV3 Core Set** je namenjen učiteljem in učencem, torej za učenje in poučevanje programiranja s pomočjo LM kompletov. Kocko programiramo s pomočjo posebnega licenčnega programskega okolja LM Education EV3, ki ga kupimo ločeno od kompleta. Slika 3 prikazuje vsebino kompleta: EV3 kocko, dva velika motorja, srednji motor, dva pritiska senzorja, barvni senzor, ultrazvočni senzor, žiro senzor, polnilno baterijo za EV3 kocko in 540 gradnikov. [7]



Slika 3: Vsebina kompleta Education EV3 Core Set [7]

Priložena plastična škatla je primerna za sortiranje in shranjevanje celotnega kompleta, kar izobraževalnim ustanovam služi predvsem za pospravljanje robotov, ko ti niso v uporabi. Navodila za uporabo in grajenje robotov so priložena, dostopna pa so tudi na svetovnem spletu in v programski opremi za programiranje kocke.

Tabela 1 prikazuje primerjavo obeh kompletov. Kocka EV3 je v obeh kompletih enaka, razlika pa nastane predvsem pri senzorjih in številu gradnikov. Pri kompletu za domačo uporabo potrebujemo še baterije za delovanje robota in daljinskega upravljalnika, pri kompletu za učence in učitelje pa polnilnik priložene baterije.

Lastnost	Home Edition	Education EV3 Core Set
inteligentna kocka	EV3	EV3
programska oprema	brezplačna – Home Edition	dodaten nakup – Education
število gradnikov	600	540
veliki motor	da – 2	da – 2
srednji motor	da – 1	da – 1



Lastnost	Home Edition	Education EV3 Core Set
pritisni senzor	da – 1	da – 2
barvni senzor	da – 1	da – 1
ultrazvočni senzor	ne	da
žiro senzor	ne	da
IR senzor + daljinski upravljalnik	da	ne
priložena baterija za kocko	ne	da
priložen polnilnik za kocko	ne	ne

Tabela 1: Primerjava kompletov Home Edition in Education EV3 Core Set

Poleg osnovnega kompleta Education obstaja še razširitveni Education EV3 Expansion Set, ki omogoča gradnjo večjih in kompleksnejših robotov. Vsebuje 850 gradnikov, ne vključuje pa nobenih dodatnih senzorjev, motorjev ali drugih elektronskih dodatkov.

### 3.3 Kocka EV3

Kocka EV3 se od svojega predhodnika NXT zelo razlikuje. Vsebuje procesor tipa ARM9 s frekvenco delovanja 300 MHz, 64 MB delovnega pomnilnika in 16 MB flash pomnilnika. Procesor poganja naložen operacijski sistem, ki temelji na Linuxovem jedru. Kocka ima 4 vhode za priklop senzorjev in 4 izhode za priklop motorjev. Za komunikacijo z računalnikom uporablja povezavo USB in omogoča priklop zunanjih USB-naprav (USB-ključev, Wi-Fi dongle itd.). Podpira tudi povezavo Bluetooth, ima režo za spominsko kartico Micro SD, šest osvetljenih gumbov in matrični monokromatski LCD-ekran z ločljivostjo  $178 \times 128$  točk. Kocke lahko s pomočjo USB ali Bluetooth povezave med seboj tudi komunicirajo in se povežejo. [8]

### 3.4 Motorji

Slika 4 prikazuje oba motorja v EV3 kompletih – veliki in srednji motor.



Slika 4: Veliki in srednji motor [9]

**Veliki motor** je močan pametni motor, ki ga lahko krmilimo na stopinjo obrata natančno. Vgrajen ima tudi senzor obratov, ki nam podaja povratno informacijo, za koliko se je motor zavrtel. Primeren je predvsem za premikanje robota ali pogonske gredi. Obrača se s hitrostjo 160–170 obratov na minuto z navorom 20 Ncm in s skrajnim navorom 40 Ncm. [9]

**Srednji motor** ima prav tako vgrajen senzor za povratno informacijo o vrtenju motorja, vendar je manj zmogljiv kot veliki motor, hkrati pa manjši in lažji, bolj odziven ter hitrejši. Primeren je za premikanje posameznih delov. Obrača se s hitrostjo 240–250 obratov na minuto z navorom 8 Ncm in s skrajnim navorom 12 Ncm. [9]

### 3.5 Senzorji

Slika 5 prikazuje senzorje, ki jih lahko uporabljamo s kocko EV3: barvni, žiro, pritiski, ultrazvočni in infrardeči senzor, infrardeči daljinski upravljalnik ter temperaturni senzor. V osnovnem kompletu EV3 so priloženi barvni, žiro, pritiski in ultrazvočni senzor. Ostali senzorji so na voljo v dodatnih kompletih ali posamično.



Slika 5: Vsi senzorji kompleta EV3 [9]

### 3.5.1 Barvni senzor

Barvni senzor (ang. Color Sensor) zaznava barvo ali intenzivnost svetlobe, ki jo sprejema skozi manjšo lečo. Senzor deluje na tri načine: barvni način, način intenzivnosti odbite svetlobe in način intenzivnosti svetlobe okolice. Frekvenca vzorčenja senzorja je 1 kHz, kar pomeni, da senzor posodablja sprejete vrednosti tisočkrat na sekundo. [9]

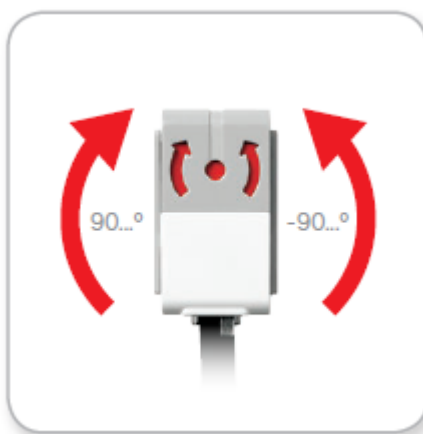
V **barvnem načinu** (ang. Color Mode) senzor prepoznava šest različnih barv: črno, modro, zeleno, rumeno, rdečo, belo in rjavo. Poleg naštetih barv pozna še t. i. No Color, s čimer zaznamuje svetlobo, ki nima izrazitih odtenkov katerekoli od prej naštetih barv. Barvni način se pogosto uporablja za vodenje robota po različnih korakih, označitev začetka oziroma konca določenega bloka in podobno. [9]

V **načinu intenzivnosti odbite svetlobe** (ang. Reflected Light Intensity Mode) modul barvnega senzorja vklopi rdečo lučko, senzor pa meri intenzivnost odbite svetlobe te lučke. Intenzivnost odbite svetlobe se razporedi na jakostni lestvici od 0 – ki označuje nič oziroma skoraj nič odbite svetlobe – do 100 – ki pomeni, da se je skoraj vsa svetloba odbila v senzor. Jakost odbite svetlobe je odvisna od bližine in svetlosti površine, od katere se svetloba odbija, kar pomeni, da je lahko senzor v tem načinu uporabljen za sledenje površini z enako intenziteto odbite svetlobe. [9]

V **načinu intenzivnosti svetlobe okolice** (ang. Ambient Light Intensity Mode) senzor meri intenzivnost svetlobe, ki vstopa skozi manjšo lečo na samem modulu senzorja. Intenzivnost sprejete svetlobe se razporedi na jakostni lestvici od 0, ki označuje nič oziroma skoraj nič sprejete svetlobe, do 100, ki pomeni, da je senzor sprejel zelo svetlo svetlobo. Posebnost tega načina delovanja je kalibracija, ki jo lahko opravimo za pravilno delovanje. Pogost način uporabe tega senzorja je izdelava in programiranje pametne budilke, ki prične zvoniti ob sprejemu sončne svetlobe s senzorja. [9]

### 3.5.2 Žiro senzor

Žiro senzor zaznava gibanje v obliki vrtenja na eni osi. Kot je razvidno iz spodnje slike (Slika 6), ima modul senzorja označen center osi, okoli njega pa so označene puščice, ki kažejo smeri, v katerih senzor prepoznava gibanje. Senzor gibanje zapisuje v merski enoti stopinje na sekundo (najvišja hitrost, ki jo senzor prepoznava, je 440 stopinj na sekundo, kar je približno 1,22 obrata na sekundo), hkrati pa beleži tudi stopinje naklona (natančnost beleženja stopinj je  $\pm 3^\circ$  na  $90^\circ$ ). Pogosti primeri uporabe so določanje, za koliko se naj določen modul robota obrne, in beleženje smeri robota. [9]



Slika 6: Označba centra osi in smeri merjenja gibanja na žiro senzorju [9]

Posebna lastnost žiro senzorja je kalibracija, ki jo je treba opraviti ob povezavi senzorja s kocko, priporočljiva pa je tudi ob vsakem zagonu programa, saj se v nasprotnem primeru lahko pojavijo napake v delovanju samega programa.

### 3.5.3 Pritisni senzor

Pritisni senzor (ang. Touch Sensor) zaznava pritisk in sprostitev tipke, ki je nameščena na modulu tega senzorja. Senzor ima tri načine prepoznavanja: pritisk, sprostitev in tapnjenje (zaporedna pritisk in sprostitev). [9]

### 3.5.4 Ultrazvočni senzor

Ultrazvočni senzor (ang. Ultrasonic Sensor) zaznava oddaljenost od površine (predmeta), ki je pred njim. Med svojim delovanjem senzor pošilja signale v obliki visokofrekvenčnih zvočnih valov, ki so za človeško uho neslišni, nato pa sprejema odboj teh valov in meri čas od trenutka oddaje do trenutka sprejema signala. Senzor vrača vrednosti v centimetrih ali inčih. Območje delovanja senzorja je od 3 cm (oz. 1 in) do 250 cm (oz. 99 in) z natančnostjo 1 cm (oz. 0,394 in). Če senzor prikazuje vrednost 255 cm (oz. 100 in), to označuje, da senzor ne zaznava nobene površine. Senzor pozna dva načina delovanja: meritveni način in način prisotnosti. Način delovanja označuje tudi lučka, ki je nameščena na modul senzorja, in utripa, če je senzor v načinu prisotnosti, oziroma sveti, če je vključen meritveni način. [9]

V **meritvenem načinu** (ang. Measure Mode) senzor meri oddaljenost površine, ki jo zaznava, kot opisano v prejšnjem odstavku.

V **načinu prisotnosti** (ang. Presence Mode) senzor zaznava signale drugih ultrazvočnih senzorjev, ki so v bližini. V tem načinu senzor signalov ne pošilja, temveč jih zgolj sprejema oziroma posluša.

Pogosti primeri uporabe ultrazvočnega senzorja so zaznavanje bližine predmeta in preprečitev trka, sledenje premikajočemu se predmetu in zaznavanje drugega robota.

### 3.5.5 Infrardeči senzor in infrardeči daljinski upravljalnik

Infrardeči senzor (ang. Infrared Sensor) zaznava infrardeče signale, ki jih oddaja in se odbijajo od drugih površin, ali infrardeče signale, ki jih oddaja infrardeči daljinski upravljalnik (ang. Remote Infrared Beacon). Infrardeči senzor pozna tri načine delovanja: način bližine, svetilnika in daljinskega upravljanja. [9]

V **načinu bližine** (ang. Proximity Mode) modul senzorja oddaja infrardeče signale, ki so za človeško oko nevidni, in sprejema odboj teh signalov. Modul glede na čas odboja računa oddaljenost predmeta in ga vrača v obliki skale od 0 (zelo blizu) do 100 (daleč). Senzor zaznava predmete, ki so oddaljeni do 70 cm, natančnost pa je nižja kot pri ultrazvočnem senzorju. [9]

V **načinu svetilnika** (ang. Beacon Mode) infrardeči senzor sprejema signale, ki jih oddaja infrardeči daljinski upravljalnik, ta pa prevzame vlogo svetilnika. V našem programu določimo kanal, ki naj ga modul infrardečega senzorja sprejema. Senzor sprejema signale, ki so oddaljeni do 200 cm v svoji smeri. Ko senzor sprejme signal, ki pripada izbranemu kanalu, oceni približno smer, iz katere signal prihaja (ang. heading), in približno razdaljo (ang. proximity) do svetilnika. Vrednost smeri podaja glede na skalo od -25 do 25, kjer 0 pomeni, da je senzor usmerjen točno proti signalu. Oddaljenost oceni z lestvico od 0 do 100. S pomočjo teh informacij lahko robot sledi svetilniku ali pa se mu izogiba. [9]

### **3.5.6 Infrardeči daljinski upravljalnik**

Infrardeči daljinski upravljalnik (ang. Remote Infrared Beacon) je posebna enota, ki se jo lahko vgradi v robota ali pa se jo uporablja iz roke. Na modulu daljinskega upravljalnika je 5 gumbov: gumb za vklop/izklop in 4 funkcijski gumbi. Funkcijski gumbi so programabilni, možne pa so tudi kombinacije dveh gumbov. Infrardeči daljinski upravljalnik ima dva načina delovanja: način svetilnika in daljinskega upravljalnika. [9]

V **načinu svetilnika** (ang. Beacon Mode) modul daljinskega upravljalnika opravlja vlogo svetilnika, ki je opisana v poglavju 3.5.5.

V **načinu daljinskega upravljalnika** (ang. Remote Mode) z modulom (oziroma gumbi na modulu) daljinsko upravljamo robot oziroma funkcije, ki smo jih definirali v našem programu. Sprejemamo lahko pritisk enega gumba ali kombinacije dveh gumbov.

### **3.5.7 Temperaturni senzor**

Temperaturni senzor (ang. Temperature Sensor) meri temperaturo na konici kovinske sonde. Senzor meri temperature od -20 °C do 120 °C (oz. od -4 °F do 248 °F) z natančnostjo 0,1 °C. Senzor se pogosto uporablja za beleženje vrednosti in spremljanje sprememb v tekočinah, za kar je posebej uporabna 6,4-cm dolga sonda na modulu senzorja. Priporočljivo je, da ga s kocko povežemo s 50-cm kablom in se na ta način izognemo morebitnim poškodbam kocke. [9]

## **3.6 Programska oprema EV3**

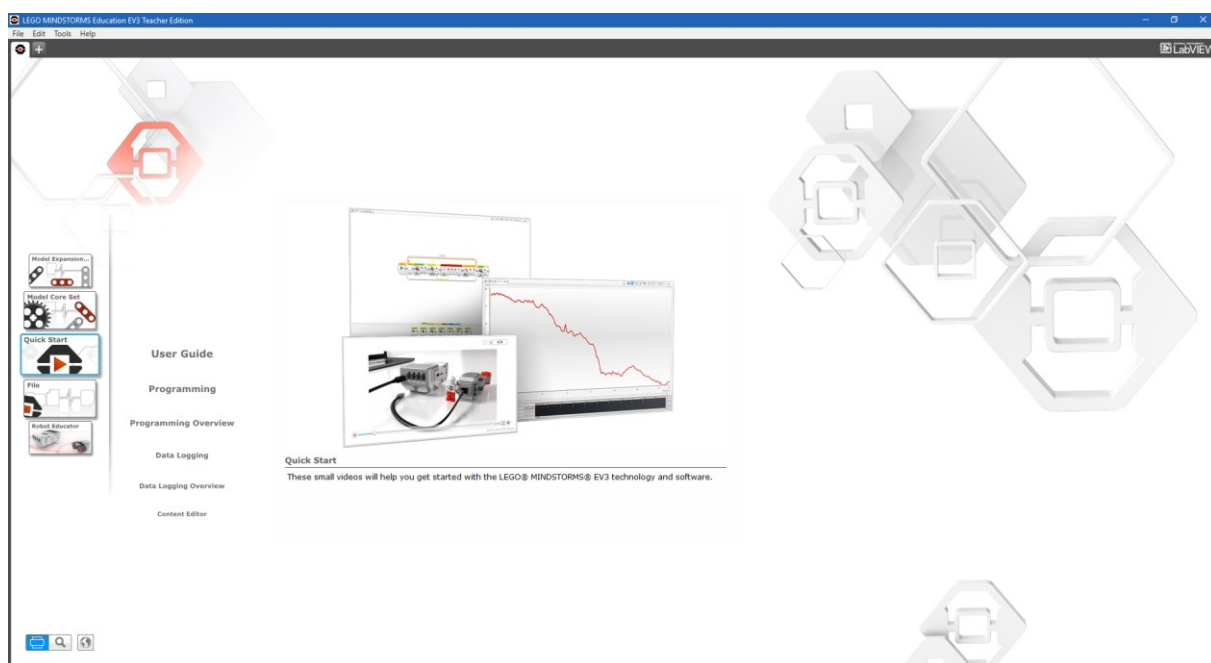
Programska oprema EV3 je namenjena enostavnemu programiranju kocke EV3. Omogoča nam vizualen in intuitiven uvod v programiranje ter beleženje podatkov, ki jih pridobivamo s pomočjo senzorjev iz kompleta EV3.

Poznamo dve izdaji programskega kompleta EV3 – različico za učence (ang. Student Edition) in učitelje (ang. Teacher Edition). V različici za učitelje so dodane informacije, viri in orodja, ki so lahko uporabni pri organizaciji in izvedbi pouka s pomočjo EV3 kompletov. Za učitelje je priporočljivo, da si naložijo različico za učitelje.

Programska oprema EV3 je na voljo za osebne računalnike z naloženim operacijskim sistemom Windows ali OSX. Za programiranje EV3 kock je na voljo tudi aplikacija EV3 Programmer App, ki je prilagojena za naprave z operacijskim sistemom Android ali iOS.

### 3.6.1 Pregled osnovnih funkcij programske opreme EV3

Slika 7 prikazuje osnovni pogled programske opreme EV3, ki se prikaže, ko zaženemo programsko opremo.



Slika 7: Osnovni pogled programske opreme EV3

Iz menija na levi strani lahko izberemo eno izmed petih glavnih funkcij programske opreme: hitri začetek (ang. Quick Start), navodila za osnovni set (ang. Model Core Set), navodila za razširitveni set (ang. Model Expansion Set), delo z datotekami in projekti (ang. File) ter vodič po robotu (ang. Robot Educator).

S pomočjo funkcije **Quick Start** si lahko ogledamo hitra navodila in napotke za uporabo programske opreme EV3.

S pomočjo funkcij **Model Core Set** in **Model Expansion Set** si lahko ogledamo navodila za izgradnjo različnih oblik robotov, ki jih lahko zgradimo s kosi iz osnovnega modela (Model Core Set) ali z dodanim razširitvenim paketom (Model Expansion Set).

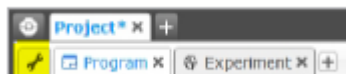
S funkcijo **File** upravljamo datoteke oziroma projekte – ustvarimo nov projekt ali pa odpremo in nato urejamo starega.

Funkcija **Robot Educator** ponuja navodila v obliki videoposnetkov za uporabo programske opreme EV3, programske opreme kocke EV3 in navodila za gradnjo robotov ter sistemov.

Z nalaganjem dodatnih programskih paketov, ki so dostopni na povezavi <https://education.lego.com/en-us/downloads/mindstorms-ev3>, lahko v programsko opremo vnesemo dodatne funkcionalnosti, kot so **Design Engineering Projects**, ki vsebuje navodila in napotke za gradnjo kompleksnejših robotov, ter **Science**, ki vsebuje navodila in napotke za znanstvene eksperimente.

V programski opremi EV3 ustvarjamo projekte, ki so sestavljeni iz programov, slik, zvokov, blokov (funkcij), spremenljivk in ostalih elementov, ki jih potrebujemo za delovanje ter usmerjanje našega robota.

Ko odpremo nov projekt, se nam prikaže okolje za programiranje novega programa. S klikom na ikono orodja, ki je na spodnji sliki (Slika 1) označena z rumeno barvo, dostopamo do lastnosti projekta in virov, ki pripadajo projektu.



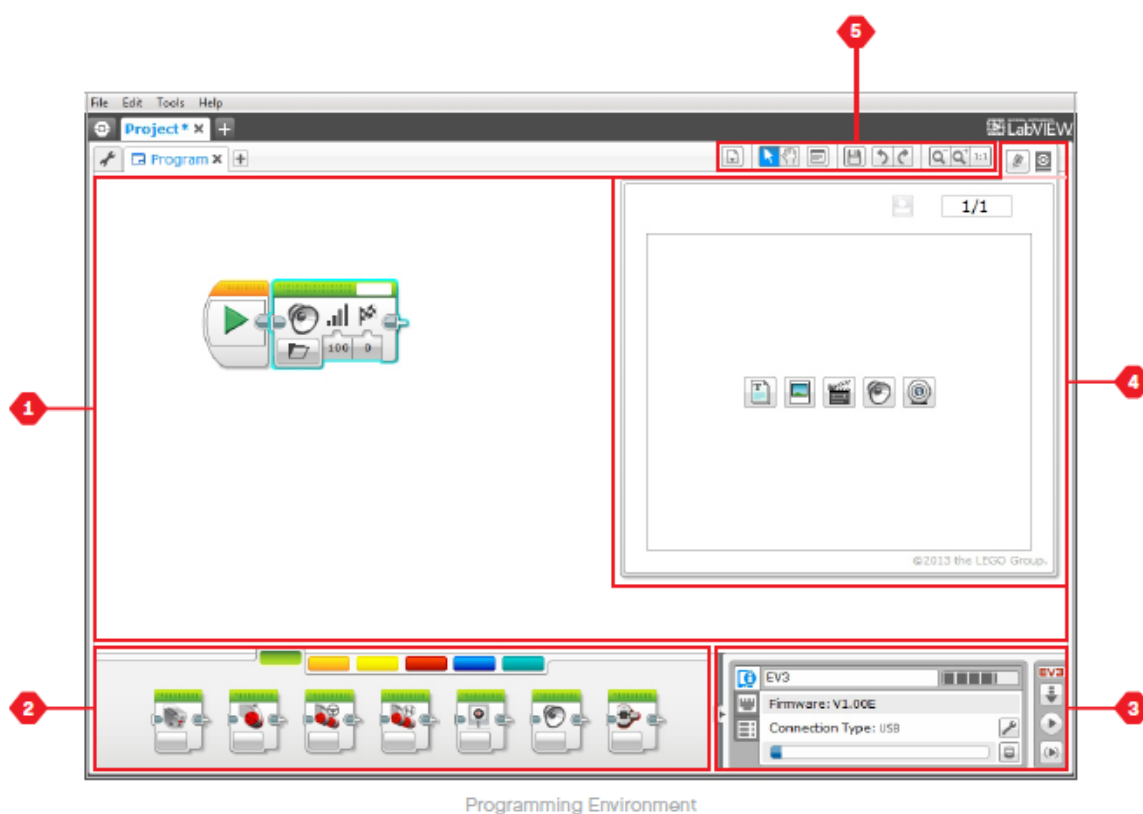
Slika 8: Ikona orodja za urejanje projekta

Projekt je lahko sestavljen iz več programov. Program je celota ukazov, ki jih robot zaporedno izpolnjuje, in vrača rezultate. V programski opremi EV3 programe gradimo grafično s pomočjo t. i. blokov (ukazov).



Kot je razvidno iz spodnje slike (Slika 9), je grafično okolje za grajenje programa sestavljeno iz več področij:

1. površina za grajenje programa (ang. Programming Canvas),
2. paleta za programiranje (ang. Programming Palettes),
3. pregled strojne opreme (ang. Hardware Page),
4. urejanje vsebine (ang. Content Editor),
5. orodna vrstica za programiranje (ang. Programming Toolbar).



Slika 9: Pregled grafičnega okolja za grajenje programa

Programe gradimo s pomočjo blokov, ki jih najdemo v paleti za programiranje in so razdeljeni na zavihke: akcijski bloki (ang. Action), bloki za potek programa (ang. Flow Control), bloki za senzorje (ang. Sensor), bloki za podatkovne operacije (ang. Data Operations), napredni bloki (ang. Advanced) in bloki po meri (ang. My Blocks).

V predelu za pregled strojne opreme je razvidno, katere kocke so trenutno povezane na računalnik, osnovne informacije o teh kockah in kateri moduli (motorji, senzorji) so povezani na posamezno kocko, njihova trenutna stanja in vrednosti.

V področju za urejanje vsebine lahko poleg grajenja programa oblikujemo tudi ustrezno dokumentacijo, navodila, opombe in beležke. Področje za urejanje vsebine je za učitelje še posebej uporabno, saj si lahko tu zabeležijo pomembne opombe za potek pouka, te informacije pa pozneje za učence ali dijake prikrijejo.

### 3.6.2 Programiranje z bloki

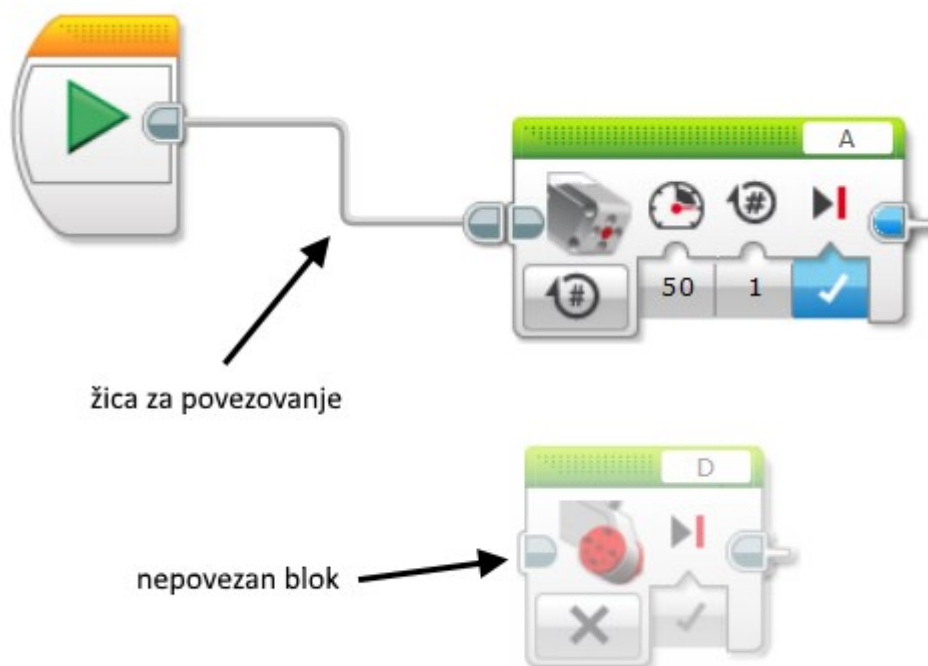
Kot že omenjeno, programe gradimo s t. i. bloki, ki jih najdemo v paleti za programiranje.

Če želimo uporabiti blok, ga izberemo iz palete in povlečemo na površino za grajenje programov. Vsak program se začne z začetnim blokom Start, ki ga prikazuje Slika 10.



Slika 10: Začetni blok Start

Bloke postavljamo enega poleg drugega ali pa jih povezujemo s pomočjo t. i. žice za povezovanje, ki jo prikazuje Slika 11. Iz slike je razvidno, da je blok, ki je nepovezan, označen oziroma siv. Takšen blok se ob izvajanju programa ne bo izvedel.



Slika 11: Bloke lahko povezujemo s t. i. žico za povezovanje. Blok, ki ni povezan, je označen (siv).

Vsi bloki, razen blokov za potek programa, imajo podobno sestavo in podoben način interakcije. Slika 12 prikazuje štiri bloke.

Pri vsakem bloku lahko vidimo priključek za vstopno in izstopno povezavo, s pomočjo katerih povežemo blok na predhodnika in naslednika.

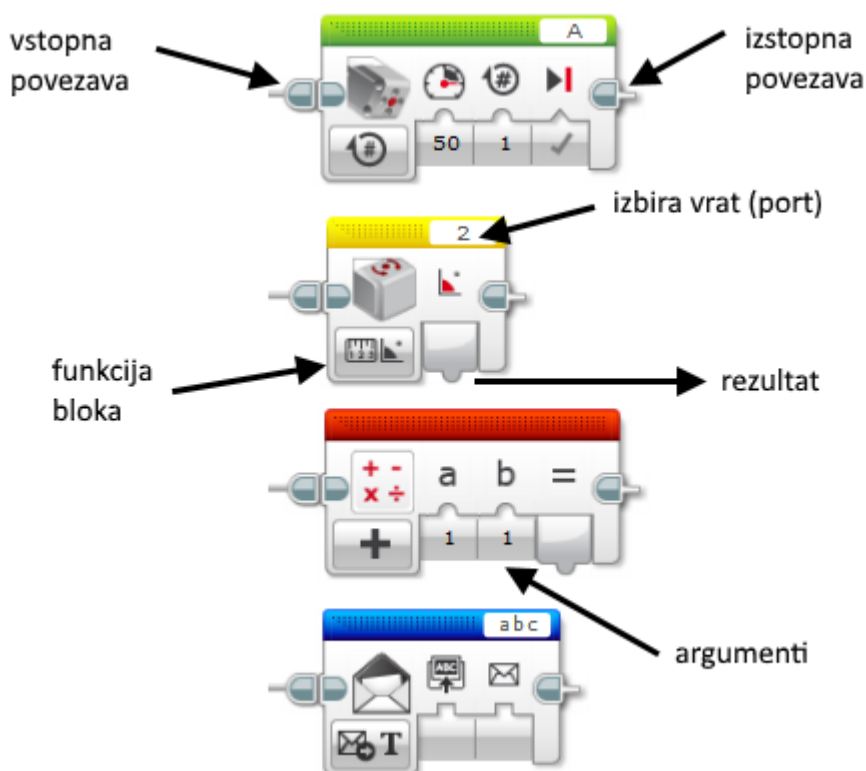
Vsi bloki, ki imajo opravka s senzorji ali motorji, imajo zgoraj desno gumb za izbiro vrat (port ali priključek). S to izbiro povemo programu (oziroma kocki), kateri senzor oziroma motor želimo upravljati s tem blokom.

Večina blokov ima spodaj levo gumb za izbiro funkcije. Pri motorjih so običajni primeri funkcij vklop, izklop, premik za določeno število obratov, premik za določeno število sekund itd. Pri senzorjih so običajno na voljo funkcije, ki so za posamezne senzorje opisane v poglavju 3.5 Senzorji.

Bloki, ki vračajo rezultate (bloki za senzorje, bloki za podatkovne operacije ipd.), imajo običajno spodaj desno polje za rezultat, ki ga lahko s pomočjo tehnike povezovanja podatkov povežemo z drugimi polji.

Bloki, ki sprejemajo vrednosti kot vhodne argumente, imajo običajno spodaj desno polja za vpis argumentov, ki jih prav tako lahko povežemo z drugimi podatki, kot je opisano v prejšnjem odstavku.

Bloki lahko sprejemajo vhodne argumente in vračajo izhodne rezultate, kakor je to običajno za funkcije v višjenivojskih programskih jezikih.

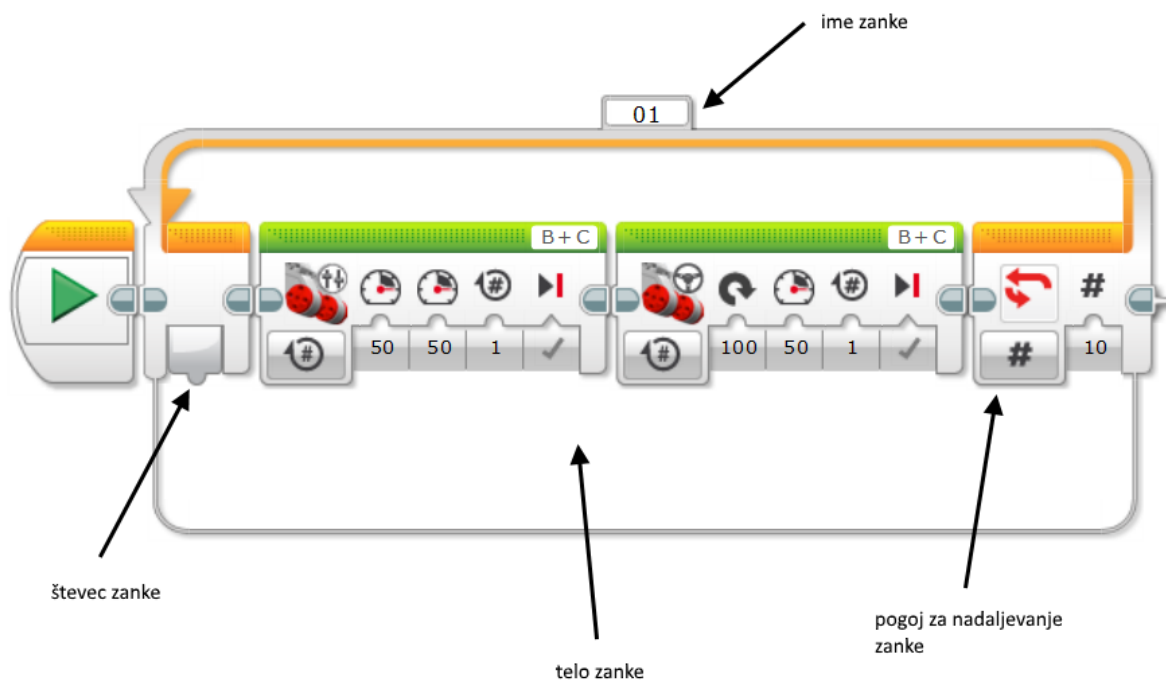


Slika 12: Bloki imajo podobno sestavo in podoben način interakcije. Na sliki so (od vrha navzdol): akcijski blok, blok za senzorje, blok za podatkovne operacije in napredni blok.

V zavihku **blokov za potek programa** so na voljo bloki, s katerimi usmerjamo in nadzorujemo potek programa. Poleg začetnega bloka (Start), s katerim začnemo vsak program, in bloka za čakanje (Wait), s katerim zakasnimo naslednji ukaz, sta tukaj še bloka Loop in Switch, ki bi jih v višjenivojskih programskih jezikih imenovali zanka in vejitev (ali while in if), ter blok za prekinitev zanke (ang. Loop Interrupt), ki bi ga v višjenivojskih programskih jezikih imenovali break.

**Zanka** (ang. Loop) je blok, ki ga prikazuje Slika 13 in je sestavljen iz polja za ime zanke, števca zanke, telesa zanke in polja za nadaljevanje zanke. S pomočjo imena zanke v nadaljevanju v bloku za prekinitev določimo, katero zanko naj ta blok prekine. Števec zanke lahko s pomočjo povezovanja podatkov povežemo z drugimi bloki. V telo zanke dodajamo bloke, ki se bodo

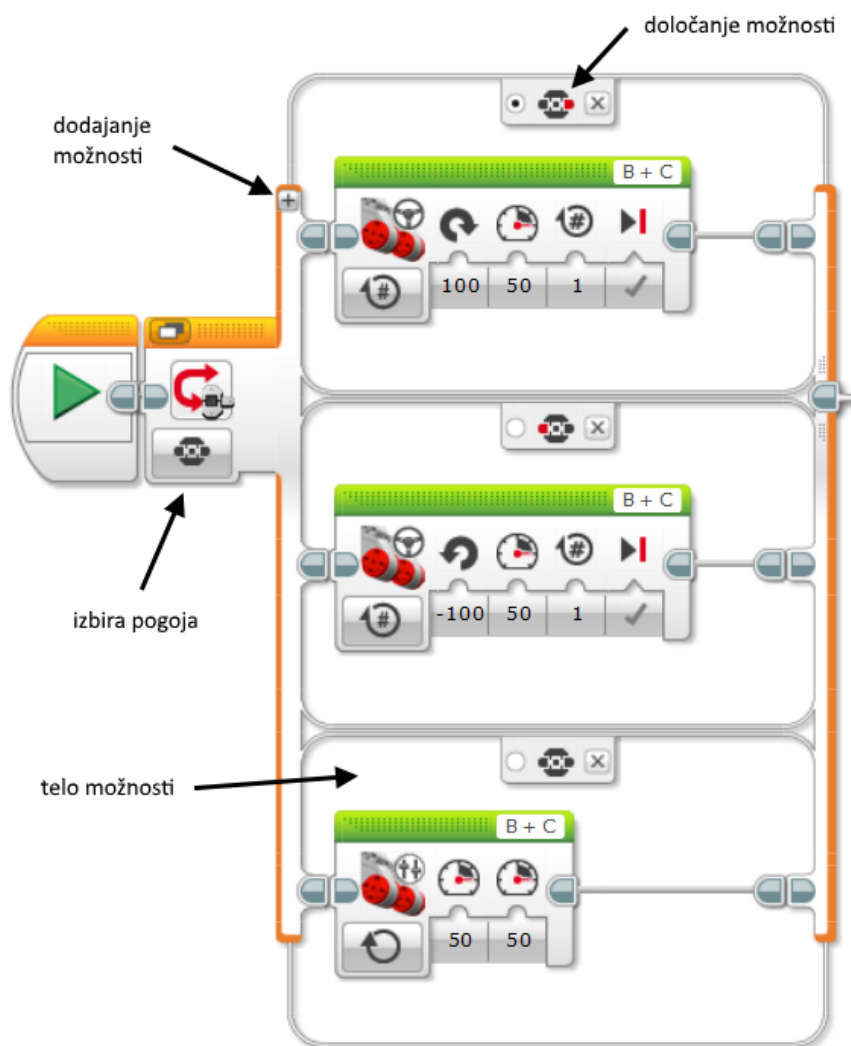
izvajali znotraj zanke, pogoj za nadaljevanje zanke pa določa, do kdaj naj se zanka izvaja. Pogoj je lahko določen kot neskončen (neskončna zanka) ali matematično, lahko ga povežemo s podatki iz drugih blokov ali pa ga določimo s pomočjo senzorjev in motorjev. Blok zanke se v izvajanju programa ponavlja, dokler je pogoj za nadaljevanje zanke izpolnjen. Zanke lahko tudi gnezdimo (ustvarjamo zanko v zanki), izvajanje gnezdenih zank pa lahko tudi pogojujemo.



Slika 13: Blok zanke je sestavljen iz več elementov

Program, ki ga prikazuje Slika 13, je sestavljen iz zanke, ki se ponovi desetkrat, znotraj zanke pa robot ponavlja naslednji postopek: najprej potuje en obrat motorjev naprej, nato pa za en obrat motorjev obrne v desno.

Slika 14 prikazuje blok **vejitev** (ang. Switch). Kot je razvidno iz slike, je blok sestavljen iz gumba za izbiro pogoja in področij za možnosti. Možnosti lahko dodajamo poljubno mnogo, znotraj vsake možnosti pa lahko dodajamo poljubno mnogo blokov, ki se bodo izvedli, če je ta možnost prava glede na pogoj. S krogcem ob možnosti določimo tudi privzet pogoj (Default), ki določa možnost, ki se bo izvedla v primeru, da nobena od možnosti ne velja. Vejitve lahko tudi gnezdimo (ustvarjamo vejitve znotraj vejitve), pogoje gnezdenih vejitev pa lahko tudi povezujemo.



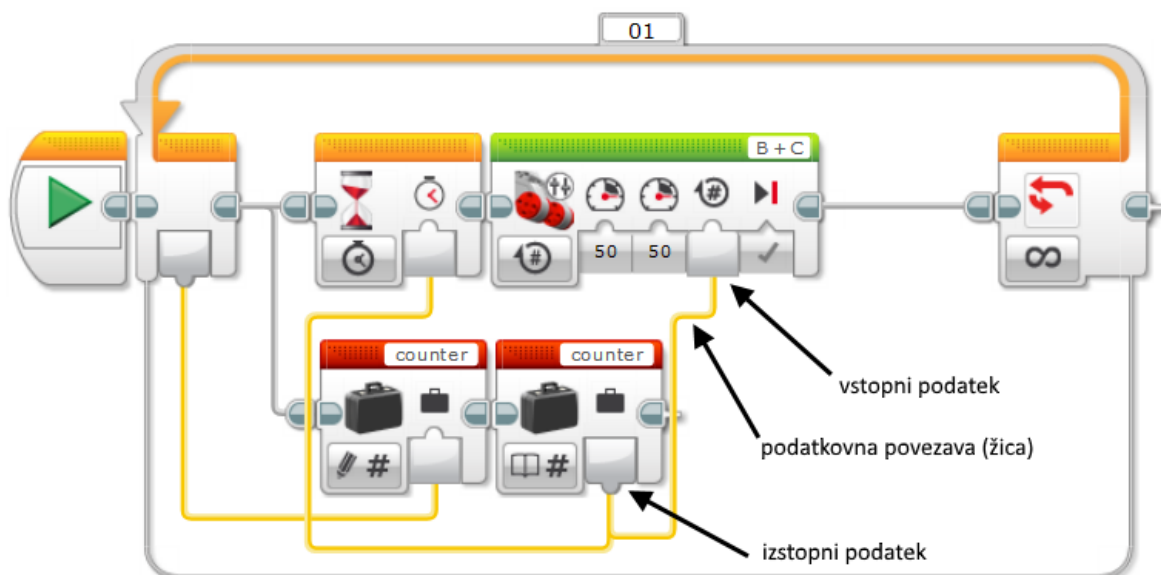
Slika 14: Blok vejitev je sestavljen iz več elementov.

Program, ki ga prikazuje Slika 14, je sestavljen iz vejitve, katere pogoj je stanje gumbov na kocki. Podane so tri možnosti: pritisk gumba desno, pritisk gumba levo in možnost, ko ni stisnjen noben gumb. Če je pritisnjen gumb desno, bo robot zavil v desno, če je pritisnjen gumb levo, pa bo robot zavil v levo. Če ni pritisnjen noben gumb, bo robot peljal naprej.

Vstopne in izstopne podatke (argumente) blokov lahko med seboj tudi povezujemo in na ta način povežemo delovanje programa.

Kot prikazuje Slika 15, lahko izstopne podatke (podatke, ki nam jih določeni bloki vračajo oz. sporočajo) povezujemo s polji za vstopne podatke (vhodnimi argumenti) blokov. Vstopne in izstopne podatke ločimo že po obliki kvadratika – izstopni imajo ušesek obrnjen navzdol,

vstopni pa navzgor. Izstopne podatke lahko tako kot vstopne povezujemo na več različnih blokov in jih tako uporabimo večkrat ter na več mestih.



Slika 15: Podatke lahko povezujemo

Slika 15 prikazuje program, ki je sestavljen iz neskončne zanke. V zanki je števec zanke (ki se povečuje z vsako ponovitvijo zanke) povezan kot vstopni podatek spremenljivke z imenom counter, vrednost te spremenljivke pa je naprej povezana kot vstopni podatek bloka Wait in bloka za premik obeh motorjev za določeno število obratov. V vsaki ponovitvi zanke se vrednost števca shrani v spremenljivko, hkrati pa program počaka toliko sekund, kolikor je trenutna vrednost spremenljivke counter, nato pa se za to enako vrednost obratov zavrtita oba motorja robota.

## **Poglavje 4      Učne priprave izbirnega predmeta Informatika**

Učne priprave izbirnega predmeta Informatika v gimnaziji, ki so osrednji del te diplomske naloge, so osredotočene predvsem na osnove programiranja. Dijak naj bi v sklopu tega predmeta osvojil osnove programiranja, programskih jezikov in algoritmov. Osrednji del predmeta predstavlja programiranje robotov LM. S tem dijaki spoznajo osnovne enote programiranja (spremenljivke, vejitve, zanke) in nekaj enostavnih algoritmov. Poleg že omenjenih področij je v učnem načrtu še nekaj prostega prostora, v katerega lahko vključimo različne snovi ali področja, uporabna za nadaljnje izobraževanje dijakov. Poleg programiranja se običajno v predmet vključi še izdelavo spletnih strani, kjer dijaki spoznajo osnove označevalnega jezika HTML in slogovnega jezika CSS, osnove obdelave fotografij in grafik ter zvoka in videa ter vključitev multimedijskih vsebin v spletne strani. Ko dijaki spoznajo osnove programiranja, se jim lahko predstavi tudi zanimiv način izdelave mobilnih aplikacij za mobilni operacijski sistem Android s spletno platformo MIT App Inventor 2. Vse omenjene snovi se lahko vključijo in povežejo v celoto v obliki zaključne seminarske naloge.

Učni načrt je razdeljen na več sklopov:

1. uvod (6 pedagoških ur), v katerem se dijakom predstavi osnove programiranja, programskih jezikov in algoritmov,
2. programiranje z roboti LM (30 pedagoških ur), v katerem dijaki sestavljajo različne robote in osvojijo uporabo programske opreme EV3 ter programiranje robotov z uporabo motorjev in senzorjev,
3. teoretično preverjanje znanja osnov programiranja in programiranja z roboti LM (2 pedagoški uri),
4. osnove spletnih strani (8 pedagoških ur), v katerem dijaki spoznajo HTML, CSS in Wordpress,
5. osnove obdelave zvočnih posnetkov (4 pedagoške ure), v katerem se dijaki spoznajo s programsko opremo Audacity,



6. programiranje mobilnih aplikacij (4–8 pedagoških ur), v katerem se dijaki spoznajo s spletno platformo za programiranje mobilnih aplikacij za mobilni operacijskih sistem Android in povezovanje omenjenega operacijskega sistema z roboti LM,
7. izdelava in predstavitev zaključnih projektnih nalog (12–16 pedagoških ur), v katerem dijaki v skupinah izdelajo projekte z roboti LM, v katere poskusijo vključiti čim več različnih tehnik in tehnologij, ki so jih spoznali med poukom.

Vsi sklopi učnega načrta so podprti z izzivi (nalogami), s katerimi dijaki teoretično znanje prenašajo na praktične probleme.

Ocenjevanje znanja poteka sprotno skozi celo leto z vrednotenjem izzivov, ki jih postavljamo dijakom. Poleg celoletnega ocenjevanja je v učnem načrtu predvidena še ocena iz teoretičnega preverjanja znanja, ki naj bi nastopilo proti koncu prvega ocenjevalnega obdobja (decembra ali v začetku januarja), in ocena zaključnega projekta, ki ga dijaki izdelajo ob koncu šolskega leta.

V učnih pripravah je nanizanih tudi nekaj primerov za izzive in naloge, ki jih podajamo dijakom. Več vaj, izzivov in nalog najdemo na spletu (npr. <http://ev3lessons.com/>).

V nadaljevanju se bo diplomska naloga navezovala predvsem na sklop uvodnih ur programiranja in programiranja robotov LM. V prilogi 1 so zbrane učne priprave, ki se navezujejo na omenjena sklopa.

Vse učne priprave predstavljajo 2 pedagoški uri, ker se predmet izvaja po blok urah.

## 4.1 Potrebščine za izvajanje predmeta

Za izvajanja predmeta po pripravljenih učnih pripravah potrebujemo:

- zadostno število osnovnih kompletov LM (vsaj 1 komplet na 4 dijake),
- vsaj 1–2 LM razširitvena kompleta (priporočljivo),
- računalniško učilnico z zadostnim številom osebnih računalnikov ali zadostno število prenosnih računalnikov (vsaj toliko, kolikor je osnovnih kompletov LM, in eden dodatni za učitelja),
- projektor,

- priporočljivo uporabo spletne učilnice (Moodle ipd.).

Vsa snov, ki je zajeta v učnih pripravah v nadaljevanju, je bila dijakom podana tudi v obliki prosojnic v spletni učilnici. Na ta način so si lahko dijaki teoretične razlage in praktične primere individualno pregledali tudi v svojem prostem času ali med reševanjem svojih programskih problemov.

## 4.2 Uvod v programiranje

Učni načrt se navezuje predvsem na programiranje, zato je smiselno, da se predmet prične z osnovami programiranja. V prilogi 1 se na ta sklop navezujejo priprave 1–3.

### Učna priprava 1

V prvih dveh urah se dijakom na kratko razloži pojme programiranje, programski jezik, program in algoritem, ki so jih dijaki sicer že spoznali pri predmetu Informatika v prvem letniku, zato dodamo še nekaj podrobnejše razlage (npr. pri programskih jezikih razložimo tudi razliko med nizkonivojskimi in visokonivojskimi programskimi jeziki in nanizamo nekaj primerov enih in drugih). Omenimo pomembnost dobrega delovanja in optimizacije algoritmov in programov ter se na kratko navežemo na t. i. O-notacijo (red velikosti ocene časovne zahtevnosti algoritma), ki jo enostavno razložimo s primerom t. i. brute force in algoritmične rešitve (npr. razvrščanje elementov). Nato preidemo na najpogostejše načine optimizacije algoritmov: deli in vladaj ter rekurzijo.

### Učna priprava 2

V naslednjih dveh urah na kratko ponovimo in osvežimo načine optimizacije algoritmov, ki smo jih omenili na koncu prejšnje ure, nato pa dijakom ponudimo nekaj problemov in ugank, ki se navezujejo na te probleme. Veliko število primernih nalog lahko najdemo v naboru nalog za računalniško tekmovanje Bober (<http://tekmovanja.acm.si/bober/naloge-re%C5%A1itve>), na spletnem portalu z ugankami Vidra (<http://vidra.si/>) ipd.

### Učna priprava 3

V zadnjih dveh urah uvodnega sklopa ponovimo diagrame poteka, ki so jih dijaki sicer spoznali že v prvem letniku. Z enostavnimi primeri programov dijake spodbudimo k razmišljanju, kaj pomeni odločitev (if), kaj pomeni zanka ter kdaj jo uporabljamo, kaj je vstopni pogoj, izstopni pogoj itd. V nadaljevanju se navežemo še na spremenljivke in različne tipe podatkov, ki bi jih shranjevali v spremenljivke. S tem zaključimo prvi sklop predmeta.

### 4.3 Programiranje robotov LEGO Mindstorms

Na sklop programiranja robotov LM se navezujejo učne priprave 4–20, ki se nahajajo v prilogi 1.

Ovisno od števila razpoložljivih kompletov LM EV3 se dijaki v tem sklopu predmeta povežejo v pare ali smiselne skupine (3–4 dijaki). Izkazalo se je, da skupine, ki štejejo več kot 4 dijake, niso ustrezne, ker si dijaki težka smiselno in pravično razdelijo zadolžitve.

Skozi celoten proces poučevanja programiranja z roboti LM dijake spodbujamo k redoljubnemu in »čitljivemu« grajenju programov ter optimizaciji.

#### Učna priprava 4

V prvi uri tega sklopa se dijaki spoznajo z osnovnim kompletom LM EV3 in sestavijo osnovnega robota. Nato robota povežejo na osebni računalnik in prvič poženejo programsko opremo EV3 ter se spoznajo z okoljem. Dijakom predstavimo različne elemente programskega okolja, okolje za programiranje in paletu za programiranje. Na koncu druge ure izdelajo svoj prvi enostavni program, ki robotu poda ukaz za vožnjo.

#### Učna priprava 5 in učna priprava 6

V naslednjih štirih urah se dijaki spoznajo z različnimi motorji in senzorji, ki so na voljo. Dijakom na podoben način, kot so motorji in senzorji predstavljeni ter opisani v poglavjih 3.4 in 3.5, podamo osnovne podatke in lastnosti motorjev ter senzorjev, nato pa dijaki povežejo nekaj senzorjev na svojega robota. Dijake seznanimo z bloki za potek programa, ki so opisani v poglavju 3.6.2, in jim podrobneje obrazložimo delovanje bloka vejitev (Switch) ter povezavo omenjenega bloka in senzorjev (pogoji).

#### Učna priprava 7

V nadaljevanju sledita dve uri, ki sta namenjeni obravnavi gnezdenih pogojev in večopravnosti (ang. multitasking). Dijakom razložimo koncept gnezdenih pogojev z uporabo praktičnih primerov, spodbudimo jih k razmišljanju o tem, kdaj so gnezdeni pogoji smiselni in kdaj ne. Večopravnost predstavimo tudi s praktičnimi primeri iz vsakdanjih opravil z računalnikom.

### **Učna priprava 8**

Naslednji dve uri temeljita na izzivu, ki ga postavimo dijakom. Na začetku ponovimo snov iz prejšnje ure (vejitev, večopravilnost), nato pa jim predstavimo izziv, ki je opisan v učni pripravi, in jih vodimo k pravilnemu razmišljanju.

### **Učna priprava 9**

Ko dijaki osvojijo vejitev, sledita dve uri, namenjeni zankam. Predstavimo blok zanke (loop) in bloke za izstop iz zanke (loop interrupt). Snov razložimo na podoben primer, kot je opisano v poglavju 3.6.2.

### **Učna priprava 10**

Nadaljujemo z uporabo spremenljivk. Dijakom razložimo uporabo bloka za spremenljivke v programskem okolju EV3 in blokov za matematične operacije. Dijakom predstavimo koncept povezovanja podatkov, kot je to opisano v poglavju 3.6.2.

### **Učna priprava 11**

Sledita dve uri, namenjeni izzivu, s katerim bodo dijaki utrdili svoje znanje z uporabo blokov vejitve, zanke, spremenljivk in povezovanja podatkov ter večopravilnosti. Izziv je podrobneje opisan v učni pripravi.

### **Učna priprava 12**

Naslednji dve uri začnemo s ponovitvijo uporabe spremenljivk, nato pa se navežemo na seznam spremenljivk in dijakom predstavimo uporabo bloka seznam (ang. array) v programski opremi EV3. Po teoretični razlagi sledijo praktične naloge, ki so opisane v učni pripravi.

### **Učna priprava 13**

Dijaki se pobližje spoznajo s senzorji in z njihovim delovanjem. Dijakom razložimo, kako lahko v programski opremi spremljajo vrednosti, ki jih vračajo senzorji. Pogledamo si žiro senzor in njegovo posebnost – kalibracijo. Vaje navežemo še na povezovanje podatkov, tako da dijake spodbudimo h grajenju programov, v katerih morajo podatke iz senzorjev uporabiti večkrat, in sicer z uporabo spremenljivk ali tehnike povezovanja podatkov.

### **Učna priprava 14**

Pri izdelavi programov hitro naletimo na težave, ki nam jih običajno povzroči kar program sam. Ta lahko zaradi naše napake ali pa zaradi relativne napake strojne opreme ter skrajnih primerov ne deluje pravilno. Dijakom predstavimo pojem hrošča (ang. bug) ter postopek razhroščevanja (ang. debugging). Na začetku postopek ponazorimo s praktičnimi primeri brez računalnika, nato pa še s primeri v programskem okolju EV3.

### **Učna priprava 15**

Naslednji dve uri sta namenjeni izzivom, ki vključujejo vse dosedanje obdelane teme. Med reševanjem izzivov dijake spodbujamo k razhroščevanju svojih rešitev. Primeri izzivov so opisani v učni pripravi.

### **Učna priprava 16**

Sledi razlaga funkcij (metod). Dijakom predstavimo, kaj funkcije oziroma metode predstavljajo, podamo primere funkcij, ki jih dijaki že poznajo, in jih spodbudimo, da navedejo še kakšno. Predstavimo, zakaj in kje so funkcije uporabne, teorijo pa nato preusmerimo na praktične primere z uporabo blokov po meri v programski opremi EV3 (My Block).

### **Učna priprava 17**

Sledi teoretično preverjanje znanja. Če je možno, naj bo preverjanje znanja elektronsko (npr. z uporabo okolja za spletno učilnico Moodle). Učna priprava vsebuje primer preverjanja znanja v okolju Moodle. Po preverjanju znanja sledi evalvacija in razprava.

### **Učna priprava 18**

V nadaljevanju dijakom predstavimo, kako lahko svoje robote opremijo še z naprednejšimi funkcijami, kot je komunikacija med dvema robotoma. Dijake spodbudimo, da se v skupinah povežejo med sabo in skušajo sestaviti programe, ki bi robote povezali.

### **Učna priprava 19**

Sledi še zadnji izziv, ki ima vključene vse tehnike, ki smo jih spoznali skozi celoten učni proces.

## 4.4 Ideje za bogatenje pouka

Programiranje robotov LM lahko obogatimo s številnimi orodji, ki so na voljo na spletu in nam omogočajo boljši nadzor nad motorji, natančnejše kalibracije senzorjev itd. Običajno so ta orodja primernejša za napredne dijake, ki že poznajo kakšen programski jezik in jim programiranje robotov v programskem okolju EV3 ne predstavlja večjega izziva, ter za reševanje naprednejših problemov.

Za naprednejše dijake, ki že poznajo vsaj kakšen višjenivojski programski jezik, je primerno okolje **Ev3dev**. Ta predstavlja operacijski sistem, ki je zgrajen na osnovi distribucije Debian Linux. OS lahko naložimo na robota LM EV3 in s tem pridobimo možnost programiranja v enem izmed programskih jezikov, ki so na voljo (Python, JavaScript, Go, C++, C ...). Programiranje v okolju ev3dev nam nudi boljši nadzor nad motorji in senzorji, hkrati pa omogoča nalaganje programskih paketov, ki ponujajo uporabo dodatne strojne opreme (USB-naprave, tipkovnice, krmilniki itd.).

Čeprav je uporaba enostavna, zahteva poznavanje vsaj enega izmed programskih jezikov, s katerimi lahko upravljamo robota. [10]

Za dijake, ki jim osnove programiranja ne predstavljajo problemov, veselilo pa bi jih programiranje za mobilne platforme, je primerno okolje **App Inventor**. To je spletno programsko okolje, ki omogoča programiranje aplikacij za operacijski sistem Android na podoben način, kot to počnemo s programi za robote LM v programskem okolju EV3. App Inventor so razvili v podjetju Google v sodelovanju z znanstveno-tehnološko ustanovo MIT. Okolje nam ponuja uporabo različnih senzorjev, ki jih premorejo naprave Android, hkrati pa lahko s posebnimi gradniki povežemo napravo Android z robotom EV3. Na ta način lahko gradimo mobilne aplikacije za nadzor robota ali izmenjavo podatkov. [11]

Uporaba App Inventorja je enostavna, na spletu pa je dostopna tudi pomoč za uporabo.

## **Poglavje 5      Zaključek**

V okviru diplomske naloge so zbrane glavne značilnosti in lastnosti robotov LEGO Mindstorms EV3 in programske opreme EV3. Cilj naloge je zlasti predstavljala sestava učnih priprav za poučevanje izbirnega predmeta Informatika v gimnaziji.

Skozi sestavo diplomske naloge so se učne priprave spreminjale in dopolnjevale glede na izkušnje v razredu, odziv dijakov in novosti, ki prihajajo. Svet robotov LM se vsakodnevno razvija in spreminja, zato je pomembno, da so učne priprave, ki so del te diplomske naloge, zgolj okvir za pouk, ki mora biti dovolj fleksibilen, da se lahko vpeljujejo novosti ter izboljšave.

Pouk in poučevanje zahtevata konstantno izpopolnjevanje in dopolnjevanje ter sledenje novostim. V poglavju 4.4, ki vsebuje ideje za bogatenje pouka, je nanizanih zgolj nekaj trenutno dostopnih orodij, ki so smiselna za uporabo pri pouku. Toda ta orodja se spreminjajo in vsakodnevno nastajajo nova.

## Seznam slik

Slika 1: Primerjava RCX, NXT in EV3 inteligentnih kock [6] .....	8
Slika 2: Vsebina kompleta EV3 Home Edition [7] .....	9
Slika 3: Vsebina kompleta Education EV3 Core Set [7] .....	10
Slika 4: Veliki in srednji motor [9] .....	12
Slika 5: Vsi senzorji kompleta EV3 [9] .....	13
Slika 6: Označba centra osi in smeri merjenja gibanja na žiro senzorju [9] .....	14
Slika 7: Osnovni pogled programske opreme EV3 .....	17
Slika 8: Ikona orodja za urejanje projekta .....	18
Slika 9: Pregled grafičnega okolja za grajenje programa .....	19
Slika 10: Začetni blok Start.....	20
Slika 11: Bloke lahko povezujemo s t. i. žico za povezovanje. Blok, ki ni povezan, je označen (siv). .....	21
Slika 12: Bloki imajo podobno sestavo in podoben način interakcije. Na sliki so (od vrha navzdol): akcijski blok, blok za senzorje, blok za podatkovne operacije in napredni blok.....	22
Slika 13: Blok zanke je sestavljen iz več elementov .....	23
Slika 14: Blok vejitev je sestavljen iz več elementov.....	24
Slika 15: Podatke lahko povezujemo .....	25

## Seznam tabel

Tabela 1: Primerjava kompletov Home Edition in Education EV3 Core Set .....	11
---	----



## Literatura

- [1] R. Kranjc, „UČNI načrt. Program osnovna šola. Računalništvo : neobvezni izbirni predmet,“ [http://www.mizs.gov.si/fileadmin/mizs.gov.si/pageuploads/podrocje/os/devetletka/program\\_razsirjeni/Racunalnistvo\\_izbirni\\_neobvezni.pdf](http://www.mizs.gov.si/fileadmin/mizs.gov.si/pageuploads/podrocje/os/devetletka/program_razsirjeni/Racunalnistvo_izbirni_neobvezni.pdf), 2013.
- [2] R. Wechtersbach, „Učni načrt. Informatika,“ Ljubljana, 2008.
- [3] „Lego Mindstorms - Wikipedia, the free encyclopedia,“ [Elektronski]. Available: [https://en.wikipedia.org/wiki/Lego\\_Mindstorms](https://en.wikipedia.org/wiki/Lego_Mindstorms). [Poskus dostopa 9 10 2015].
- [4] „History - Mindstorms LEGO.com,“ [Elektronski]. Available: <http://www.lego.com/en-us/mindstorms/history>. [Poskus dostopa 15 08 2015].
- [5] „Lego Mindstorms EV3 - Wikipedia, the free encyclopedia,“ [Elektronski]. Available: [https://en.wikipedia.org/wiki/Lego\\_Mindstorms\\_EV3](https://en.wikipedia.org/wiki/Lego_Mindstorms_EV3). [Poskus dostopa 12 01 2016].
- [6] L. Valk, „EV3 and NXT: Difference and Compatibility,“ [Elektronski]. Available: <http://robotsquare.com/2013/07/16/ev3-nxt-compatibility/>. [Poskus dostopa 10 12 2016].
- [7] L. Valk, „Robot Square - The Difference Between LEGO MINDSTORMS EV3 Home Edition (#31313) and LEGO MINDSTORMS Education EV3 (#45544) - Robot Square,“ [Elektronski]. Available: <http://robotsquare.com/2013/11/25/difference-between-ev3-home-edition-and-education-ev3/>. [Poskus dostopa 20 12 2015].
- [8] „Comparing the NXT and EV3 bricks,“ Bot Bench, 25 11 2013. [Elektronski]. Available: <http://botbench.com/blog/2013/01/08/comparing-the-nxt-and-ev3-bricks/>. [Poskus dostopa 27 12 2015].
- [9] LEGO, „Mindstorms education EV3 User guide,“ 2016.

- [10] R. Hempel in D. Lechner, „ev3dev,“ [Elektronski]. Available: <http://www.ev3dev.org/>. [Poskus dostopa 10 12 2016].
  
- [11] Wikipedia, „Wikipedia - App Inventor for Android,“ [Elektronski]. Available: [https://en.wikipedia.org/wiki/App\\_Inventor\\_for\\_Android](https://en.wikipedia.org/wiki/App_Inventor_for_Android). [Poskus dostopa 10 12 2016].

# Priloga 1: Učne priprave

## UČNA PRIPRAVA

Predmet: **IP Informatika**

Program: gimnazija

Enota: **UVOD**

Tema: **Osnove programiranja**

Tip učne enote: kombinirana

Zap. št.  
priprave:

1

Oddelek:

Datum:

Zap. št. ure:

1-2

Šola:

Učitelj:

Šolsko leto:

Operativni vzgojno-izobraževalni cilji:

Dijaki osvojijo osnovne pojme programiranja, programskih jezikov in algoritmov ter optimizacije algoritmov. Dijaki razmišljajo o različnih načinih optimizacije.

Učne oblike: frontalna, individualna

Učne metode: razlaga, diskusija, reševanje problemov

Viri: spletni učbenik: <http://lusy.fri.uni-lj.si/ucbenik/book/index.html>

Učni  
pripomočki: tabla, flomastri različnih barv, A4 listi s številkami od 1 do 10

Didaktična struktura ure:

1. Uvod (5 min) – uvodna motivacija
2. Razlaga osnovnih pojmov (20 min) – programiranje, programski jezik itd.
3. Reševanje problema iskanja in spoznavanje različnih algoritmov (55 min)
4. Diskusija (10 min)

Učitelj:	Dijaki:
<p>Na začetku predstavi potek predmeta.</p> <p>Razloži osnovne pojme, ki se nanašajo na programiranje:</p> <ul style="list-style-type: none"> <li>• programiranje: pisanje programov</li> <li>• program: zaporedje navodil – po navadi sestavljen iz več algoritmov             <ul style="list-style-type: none"> <li>○ kako izgleda program?</li> <li>○ kakšna je njegova funkcija?</li> <li>○ zakaj jih uporabljamo?</li> <li>○ ali ga razumemo oziroma kdo ga razume?</li> </ul> </li> <li>• algoritem: zaporedje navodila za rešitev problema             <ul style="list-style-type: none"> <li>○ natančen</li> <li>○ nedvoumen</li> <li>○ dobro strukturiran</li> </ul> </li> <li>• programski jezik: jezik za programiranje oziroma podajanje navodil, ki je razumljiv stroju oziroma računalniku</li> </ul> <p>Različni programski jezik:</p> <ul style="list-style-type: none"> <li>• vsi imajo osnovne gradnike</li> <li>• vsi imajo sintaktična pravila (slovnico)</li> <li>• nivoji:             <ul style="list-style-type: none"> <li>○ strojni jezik – težavno programiranje</li> <li>○ zbirni programski jezik – programske kratice, lažje programiranje</li> <li>○ visokonivojski programski jeziki – kompleksnost ukazov, človeku bližje pisanje</li> </ul> </li> </ul> <p>Pripravimo 10 A4 listov z različnimi številkami (npr. vsak list ima eno številko od 1 do 10). Pred tablo pokličemo 10 naključnih dijakov in vsakemu damo en list (številke uredimo po velikosti). Dijaki naj imajo številke obrnjene proti sebi, da jih ostali dijaki v razredu ne vidijo. Nato dijakom damo navodila, naj najdejo številko (npr. 7). Namignemo jim, naj razmišljajo o algoritmu, ki ga bodo uporabili za iskanje. Dijake spodbujamo k čim boljšemu opisu njihovega algoritma.</p> <p>Ko izberejo prvo številko, ki jo želijo odkriti, jo povprašamo, kako so si jo izbrali. Po navadi je odgovor: »poljubno«, kar je dobro in jih napotimo, naj na ta način izbirajo, dokler ne pridejo do prave številke.</p>	<p>Poslušajo, sodelujejo, zapisujejo</p> <p>10 dijakov drži številke ostali iščejo eno številko</p> <p>razmišljajo o algoritmu za iskanje</p> <p>brute force iskanje</p>

Učitelj:	Dijaki:
<p>Ko pridejo do prave številke, jih spodbudimo k razmišljanju, kolikokrat bi morali iskati v najslabšem primeru in kolikokrat v najboljšem.</p> <p>Razložimo pojme:</p> <ul style="list-style-type: none"> <li>• brute force: postopek brez posebnega algoritma – iskanje »na srečo«</li> <li>• worse case: najslabši primer izvajanja algoritma</li> <li>• best case: najboljši primer izvajanja algoritma</li> </ul> <p>Postopoma dijake poskusimo privedi do drugih načinov iskanja:</p> <ul style="list-style-type: none"> <li>• deli in vladaj: razdelite problem na pol in iščite po eni polovici</li> <li>• rekurzija: po vsaki polovici zopet uporabite enak algoritem</li> </ul> <p>Poskusimo izvajanje z različnimi algoritmi in dijake spodbudimo k razmišljanju o časovni zahtevnosti oziroma tako imenovani O-notaciji.</p> <p>“Ko razmišljamo o hitrosti delovanja nekega postopka, po navadi bolj kot sam čas izvajanja zanima to, kako hitro čas izvajanja narašča v odvisnosti od velikosti primerka problema. Temu pravimo red časovne zahtevnosti in ga pogosto označimo z velikim <math>O</math>.”</p> <p>Učno uro zaključimo z diskusijo o smiselnosti uporabe različnih algoritmov.</p> <p>Naslednjo uro bomo poskusili različne optimizacije algoritmov uporabiti v praksi.</p>	<p>razmišljajo o worse case in best case</p> <p>skušajo problem rešiti na druge načine</p> <p>sodelujejo v diskusiji</p>

**UČNA PRIPRAVA**Predmet: **IP Informatika**

Program: gimnazija

Enota: **UVOD**Tema: **Optimizacija algoritmov**

Tip učne enote: kombinirana

Zap. št.  
priprave:

2

Oddelek:

Datum:

Zap. št. ure:

3-4

Šola:

Učitelj:

Šolsko leto:

Operativni vzgojno-izobraževalni cilji:

Dijaki že poznajo različne načine optimizacije algoritmov (deli in vladaj, rekurzija ...).

Skozi reševanje problemov dijaki spoznajo, kateri načini so uporabni v določenih situacijah.

Učne oblike: frontalna, individualna

Učne metode: razlaga, diskusija, reševanje problemov

Viri: nabori nalog

Učni

pripomočki: tabla, flomastri različnih barv, A4 listi s številkami od 1 do 10

Didaktična struktura ure:

1. Uvod (5 min) – uvodna motivacija, ponovitev snovi od prejšnje ure
2. Skupno reševanje problemov (30 min)
3. Individualno reševanje problemov (30 min)
4. Evalvacija rešitev (20 min)
5. Diskusija (5 min)

Učitelj:	Dijaki:
<p>Na začetku ponovimo snov iz prejšnje ure.</p> <p>Kratka ponovitev optimizacije:</p> <ul style="list-style-type: none"> <li>• deli in vladaj</li> <li>• rekurzija</li> </ul> <p>Skupno reševanje problemov, ki so primerni za optimizirane algoritme:</p> <ul style="list-style-type: none"> <li>• urejanje elementov,</li> <li>• iskanje elementov,</li> <li>• ...</li> </ul> <p>Dijakom lahko zopet razdelimo A4 liste s številkami, ki smo jih imeli pripravljene že na prejšnji uri. Tokrat dijake spodbudimo, da iščejo optimizirane rešitve.</p> <p>Postopoma pridemo do treh največkrat uporabljenih algoritmov za razvrščanje:</p> <ul style="list-style-type: none"> <li>• navadno izbiranje, pri katerem vedno izbiramo najmanjši element seznama in ga menjamo s trenutno izbranim elementom,</li> <li>• navadno vstavljanje, pri katerem vedno elemente vstavljamo v namišljen urejen seznam, in</li> <li>• razvrščanje z mehurčki (ang. bubble sort), pri katerem vedno primerjamo samo sosednja elementa, kar pomeni, da se vedno največja števila (kot mehurčki) zbirajo na koncu seznama.</li> </ul> <p>Z dijaki skupaj površno razmislimo o O-notaciji treh predstavljenih algoritmov s predstavitevijo števila primerjav, ki je potrebuje posamezni algoritem.</p> <p>Individualno reševanje vnaprej pripravljenih problemov.</p> <p>Evalvacija rešitev in diskusija.</p> <p>Naslednjo uro bomo različne algoritme poskusili predstaviti z diagrami poteka.</p>	<p>Poslušajo, sodelujejo, zapisujejo</p> <p>Sodelujejo.</p> <p>Rešujejo vnaprej pripravljene probleme.</p> <p>Sodelujejo v razpravi.</p>

**UČNA PRIPRAVA**Predmet: **IP Informatika**

Program: gimnazija

Enota: **UVOD**Tema: **Diagrami poteka, vejitve in zanke, spremenljivke**

Tip učne enote: kombinirana

Zap. št.  
priprave:

3

Oddelek:

Datum:

Zap. št. ure:

5-6

Šola:

Učitelj:

Šolsko leto:

Operativni vzgojno-izobraževalni cilji:

Dijaki že poznajo diagrame poteka, zato le osvežijo znanje o njih in različne algoritme, ki so jih spoznali v prejšnji uri, poskusijo prenesti na papir.

Dijaki se srečajo s pojmi: spremenljivka, vejitev (if), zanka (loop).

Učne oblike: frontalna, individualna

Učne metode: razlaga, diskusija, reševanje problemov

Viri: spletni učbenik: <http://lusy.fri.uni-lj.si/ucbenik/book/index.html>

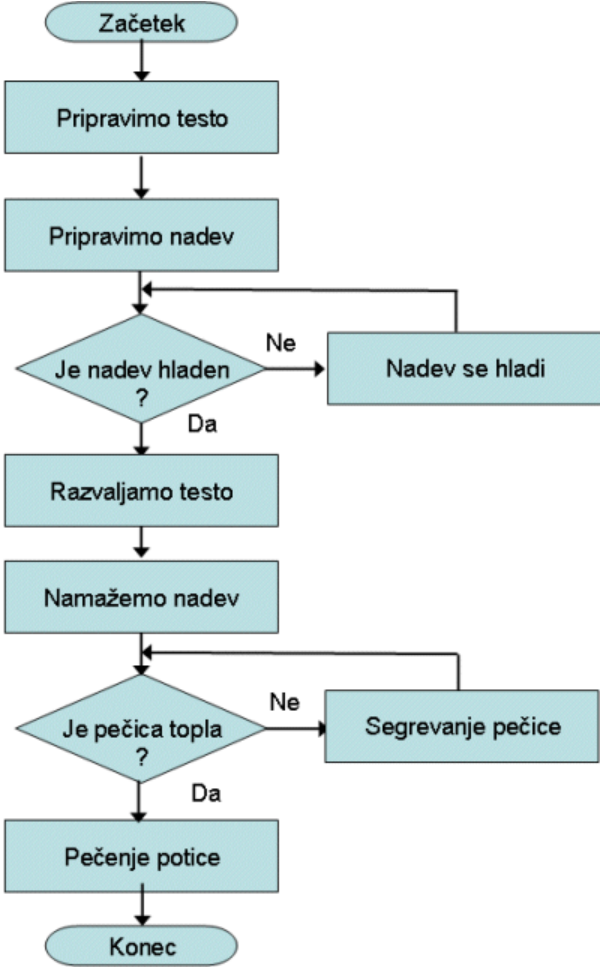
Učni

pripomočki: tabla, flomastri različnih barv

Didaktična struktura ure:

1. Uvod (5 min) – uvodna motivacija, ponovitev snovi od prejšnje ure
2. Ponovitev diagramov poteka (20 min)
3. Razlaga osnov if stavka in zanke (10 min)
4. Gradnja algoritmov za iskanje iz prejšnje ure z diagramom poteka (50 min)
5. Diskusija (5 min)



Učitelj:	Dijaki:
<p>Na začetku ponovimo snov iz prejšnje ure.</p> <p>Razloži osnove diagramov poteka:            grafični prikaz izvajanja posameznih delov programa            lažja predstava</p> <p>Gradniki diagramov poteka:</p> <ul style="list-style-type: none"> <li>• akcijski pravokotniki (ali drugi simboli)</li> <li>• krog – začetek in konec</li> <li>• romb – vnos ali izpis</li> <li>• pravokotnik (škatla) – izračun, prireditev, opravilo</li> <li>• deltoid – odločitev, vejitev</li> <li>• črte in puščice – označujejo zaporedje akcij</li> </ul> <p>Pripravimo nazoren primer diagrama poteka vsakdanjega opravila (peka potice) – dijake spodbudimo, da diagram poteka sestavijo sami, mi jim podamo samo ključne elemente.</p>  <pre> graph TD     Start([Začetek]) --&gt; PrepDough[Pripravimo testo]     PrepDough --&gt; PrepFilling[Pripravimo nadev]     PrepFilling --&gt; IsFillingCold{Je nadev hladen?}     IsFillingCold -- Ne --&gt; FillingCooling[Nadev se hladi]     FillingCooling --&gt; IsFillingCold     IsFillingCold -- Da --&gt; RollDough[Razvaljamo testo]     RollDough --&gt; SpreadFilling[Namažemo nadev]     SpreadFilling --&gt; IsOvenHot{Je pečica topla?}     IsOvenHot -- Ne --&gt; PreheatOven[Segrevanje pečice]     PreheatOven --&gt; IsOvenHot     IsOvenHot -- Da --&gt; BakePotica[Pečenje potice]     BakePotica --&gt; End([Konec])   </pre>	<p>Poslušajo, sodelujejo, zapisujejo</p> <p>Sodelujejo pri gradnji in razlagi diagrama poteka.</p>

Učitelj:	Dijaki:
<p>S pomočjo diagramov poteka razložimo pojem vejitve (if) in pojem zanke ter spremenljivke.</p> <p>Zanko poskusimo razložiti s praktičnim primerom:</p> <ul style="list-style-type: none"> <li>• naključnega dijaka prosimo, naj ponavlja določen gib</li> <li>• ostali v razredu naj ga ustavijo, ko bo stvar naredil desetkrat</li> </ul> <p>Dijake spodbudimo k razmišljanju, kakšen algoritem so izvajali.</p> <p>Postopoma pridemo do potrebe po shranjevanju podatkov. Tu nastopi razlage pojma spremenljivke, nato pa še navezava na različne tipe podatkov in sezname podatkov.</p> <p>Dijakom postavimo nekaj problemov in jih spodbudimo k reševanju s pomočjo diagramov poteka.</p> <p>Primeri:</p> <ul style="list-style-type: none"> <li>• odštevanje – vnos dveh števil, odštej manjše število od večjega</li> <li>• deljivost – vnos dveh števil, če je prvo število deljivo z drugim številom, izpiši vse večkratnike drugega števila do prvega števila</li> <li>• zvezdice – program za izpis pravokotnika zvezdic dane širine in višine</li> </ul> <p>V naslednji uri se začne delo z LEGO Mindstorms, zato naj se dijaki do naslednje ure povežejo v skupine po 2-3 dijake (odvisno od števila kompletov EV3, ki jih imamo na voljo).</p>	<p>Poslušajo, sodelujejo, zapisujejo</p> <p>Sodelujejo</p> <p>Sodelujejo, rešujejo probleme</p>

**UČNA PRIPRAVA**Predmet: **IP Informatika**

Program: gimnazija

Enota: **LEGO Mindstorms**Tema: **Uvod**

Tip učne enote: kombinirana

Zap. št.  
priprave:

4

Oddelek:

Datum:

Zap. št. ure:

7-8

Šola:

Učitelj:

Šolsko leto:

Operativni vzgojno-izobraževalni cilji:

Dijaki se spoznajo z LEGO Mindstorms roboti in programsko opremo EV3.

Učne oblike: kombinirana

Učne metode: razlaga, diskusija, reševanje problemov

Viri:

navodila za uporabnike LEGO Mindstorms EV3  
[https://le-www-live-s.legocdn.com/sc/media/files/user-guides/ev3/ev3\\_user\\_guide\\_us-ab5d4fc71211c1edc77a7362bab2d88a.pdf?la=en-us](https://le-www-live-s.legocdn.com/sc/media/files/user-guides/ev3/ev3_user_guide_us-ab5d4fc71211c1edc77a7362bab2d88a.pdf?la=en-us)

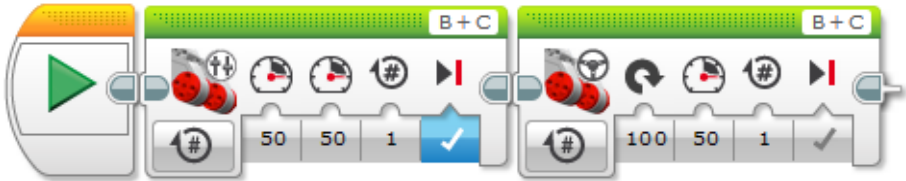
Učni

pripomočki:

tabla, flomastri različnih barv, osebni računalnik, osnovni komplet LEGO Mindstorms EV3

Didaktična struktura ure:

1. Uvod (5 min) – uvodna motivacija
2. Pregled osnovnega kompleta EV3 (20 min)
3. Sestavljanje osnovnega robota (20 min)
4. Osnove programske opreme EV3 (30 min)
5. Izdelava prvega programa (10 min)
6. Diskusija (5 min)

Učitelj:	Dijaki:
<p>Na začetku dijakom predstavimo osnovni komplet LEGO Mindstorms EV3.</p> <p>Dijaki se razdelijo v vnaprej formirane skupine. Razdelimo osnovne komplete EV3.</p> <p>Pregledamo osnovne gradnike kompleta EV3:</p> <ul style="list-style-type: none"> <li>• kocka,</li> <li>• motorji,</li> <li>• senzorji.</li> </ul> <p>Razložimo osnovne lastnosti vseh gradnikov.</p> <p>Vodimo dijake do sestavljenega robota.</p> <p>Predstavimo osnovne komponente programske opreme EV3:</p> <ul style="list-style-type: none"> <li>• navodila,</li> <li>• projekti,</li> <li>• programi,</li> <li>• površina za grajenje programa,</li> <li>• paleta za programiranje – bloki,</li> <li>• pregled strojne opreme.</li> </ul> <p>Dijake vodimo do izdelave prvega programa za premik robota. Naprednejši dijaki zgradijo tudi naprednejše programe.</p>  <p>Naslednjo uro sledi natančnejši pregled senzorjev.</p>	<p>Se razdelijo v skupine</p> <p>Poslušajo, sodelujejo, zapisujejo</p> <p>Sestavijo osnovni model robota po navodilih.</p> <p>Poslušajo, sodelujejo, zapisujejo</p> <p>Programirajo in opazujejo obnašanje robota</p>

**UČNA PRIPRAVA**Predmet: **IP Informatika**

Program: gimnazija

Enota: **LEGO Mindstorms**Tema: **Motorji in senzorji**

Tip učne enote: kombinirana

Zap. št.  
priprave:

5

Oddelek:

Datum:

Zap. št. ure:

9-10

Šola:

Učitelj:

Šolsko leto:

Operativni vzgojno-izobraževalni cilji:

Dijaki že poznajo osnovne gradnike kompleta EV3 in osnove programske opreme EV3. Dijaki natančneje spoznajo različne motorje in senzorje ter njihove lastnosti. Dijaki različne motorje in senzorje preizkusijo v praksi.

Učne oblike: kombinirana

Učne metode: razlaga, diskusija, reševanje problemov

Viri:

navodila za uporabnike LEGO Mindstorms EV3  
[https://le-www-live-s.legocdn.com/sc/media/files/user-guides/ev3/ev3\\_user\\_guide\\_us-ab5d4fc71211c1edc77a7362bab2d88a.pdf?la=en-us](https://le-www-live-s.legocdn.com/sc/media/files/user-guides/ev3/ev3_user_guide_us-ab5d4fc71211c1edc77a7362bab2d88a.pdf?la=en-us)

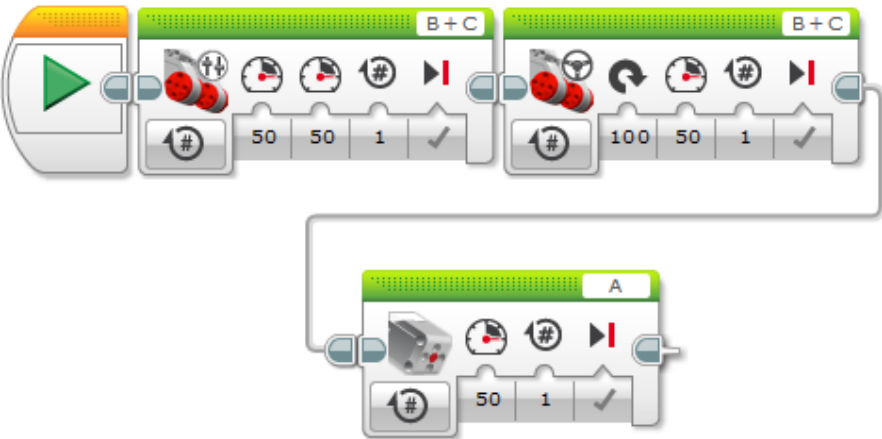
Učni

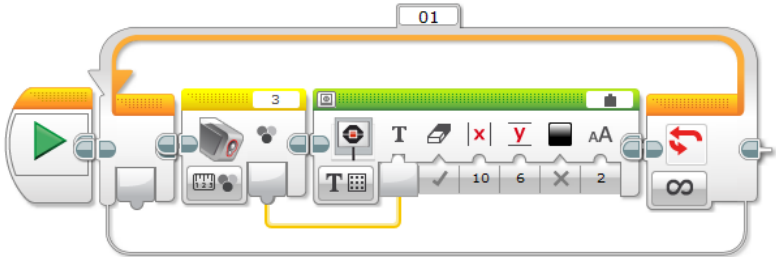
pripomočki:

tabla, flomastri različnih barv, osebni računalnik, osnovni komplet LEGO Mindstorms EV3

Didaktična struktura ure:

1. Uvod (5 min) – uvodna motivacija
2. Natančen pregled motorjev (20 min) – med pregledom dijaki povezujejo posamezne senzorje in preizkušajo njihove različne načine delovanja
3. Natančen pregled senzorjev (60 min) – med pregledom dijaki povezujejo posamezne senzorje in preizkušajo njihove različne načine delovanja
4. Diskusija (5 min)

Učitelj:	Dijaki:
<p>Na začetku ponovimo in osvežimo snov iz prejšnje ure.</p> <p>Dijakom natančneje razložimo delovanje in uporabo različnih motorjev.</p> <ul style="list-style-type: none"> <li>• veliki motor <ul style="list-style-type: none"> <li>○ močan pametni motor</li> <li>○ natančnost = 1 stopinja</li> <li>○ senzor obratov</li> <li>○ hitrost = 160-170 obratov/minuto</li> <li>○ navor = 20 Ncm – 40 Ncm</li> <li>○ premikanje robota ali pogonske gredi</li> </ul> </li> <li>• srednji motor <ul style="list-style-type: none"> <li>○ šibkejši motor</li> <li>○ bolj natančen</li> <li>○ bolj odziven</li> <li>○ hitrejši – hitrost = 240-250 obratov/minuto</li> <li>○ navor = 8 Ncm – 12 Ncm</li> <li>○ premikanje elementov (npr. roka)</li> </ul> </li> </ul> <p>Dijaki imajo že sestavljen osnovnega robota. Spodbudimo jih, da na robota povežejo še srednji motor in preizkusijo delovanje. Ob tem jih še vodimo po programski opremi EV3, da izberejo pravilna vrata itd.</p>  <p>Dijakom natančneje predstavimo različne senzorje, ki so na voljo v osnovnem kompletu EV3 in njihovo delovanje. Za lažji prikaz delovanja senzorjev vnaprej pripravimo programe, ki jih dijaki zaženejo na svojih robotih. Vsak posamezen program je izdelan tako, da na zaslonu kocke izpisuje trenutno vrednost, ki jo vrača senzor.</p>	<p>Poslušajo, sodelujejo, zapisujejo</p> <p>Na robota vgradijo srednji motor in z enostavnim programom preizkusijo delovanje</p> <p>Poslušajo, sodelujejo, zapisujejo, preizkušajo</p>

Učitelj:	Dijaki:
<p>Senzorji:</p> <ul style="list-style-type: none"> <li>barvni senzor           <ul style="list-style-type: none"> <li>zaznava barvo ali intenzivnost svetlobe</li> <li>trije načini delovanja               <ul style="list-style-type: none"> <li>barva – črna, modra, zelena, rumena, rdeča, bela, rjava in »no color«</li> <li>odbita svetloba (0-100)</li> <li>svetloba ambients (0-100)</li> </ul> </li> <li>frekvenca vzorčenja = 1kHz (1000 x/s)</li> <li>praktični preizkus vseh treh načinov in spremljanje vrednosti</li> <li>program:</li> </ul> </li> </ul>	<p>Med razlago posameznega senzorja naj dijaki tega povežejo na robota in preizkusijo različne načine delovanja</p>
	<p>Med razlago posameznega senzorja naj dijaki tega povežejo na robota in preizkusijo različne načine delovanja</p>
<ul style="list-style-type: none"> <li>žiro senzor           <ul style="list-style-type: none"> <li>zaznavanje vrtenja na eni osi</li> <li>naprava, ki jo ima vsak dijak in vsebuje žiro senzor? (mobilni telefon)</li> <li>označba centra osi in puščice</li> <li>stopinj/sekundo</li> <li>najvišja hitrost = 440 stopinj/sekundo (1,22 obrata na sekundo)</li> <li>beleženje stopinj naklona (3,33% relativna napaka)</li> <li>kalibracija</li> <li>program (ob zagonu programa je potrebno robota položiti na trdna tla in pritisniti sredinsko tipko na kocki, da se izvede kalibracija žiro senzorja)</li> </ul> </li> </ul>	<p>Med razlago posameznega senzorja naj dijaki tega povežejo na robota in preizkusijo različne načine delovanja</p>

Učitelj:	Dijaki:
<div data-bbox="264 309 1110 840" data-label="Diagram"> </div> <ul style="list-style-type: none"> <li>pritisni senzor             <ul style="list-style-type: none"> <li>zaznavanje pritiska tipke</li> <li>trije načini delovanja: pritisk, sprostitvev in tapnjenje</li> <li>program:</li> </ul> </li> </ul> <div data-bbox="277 1061 1031 1310" data-label="Diagram"> </div> <ul style="list-style-type: none"> <li>ultrazvočni senzor             <ul style="list-style-type: none"> <li>dva načina delovanja – oddaljenost in prisotnost</li> <li>oddaljenost od površine (predmeta)</li> <li>ultrazvočno valovanje</li> <li>območje delovanja 3-250 cm</li> <li>natančnost = 1 cm</li> <li>če senzor prikazuje 255, pomeni, da ne zaznava ničesar</li> <li>lučka</li> </ul> </li> </ul> <div data-bbox="277 1657 1031 1906" data-label="Diagram"> </div>	<p>Med razlago posameznega senzorja naj dijaki tega povežejo na robota in preizkusijo različne načine delovanja</p>
Naslednjo uro sledi izdelovanje programov s senzorji in pogoji.	



**UČNA PRIPRAVA**Predmet: **IP Informatika**

Program: gimnazija

Enota: **LEGO Mindstorms**Tema: **Pogoji in senzorji**

Tip učne enote: kombinirana

Zap. št.  
priprave:

6

Oddelek:

Datum:

Zap. št. ure:

11–12

Šola:

Učitelj:

Šolsko leto:

Operativni vzgojno-izobraževalni cilji:

Dijaki že poznajo senzorje, ki so del osnovnega kompleta EV3. Dijaki ponovijo osnovne lastnosti senzorjev in načine delovanja. Dijaki se spoznajo z vejitvijo (if) in pogojevanjem s senzorji.

Učne oblike: kombinirana

Učne metode: razlaga, diskusija, reševanje problemov

Viri:

navodila za uporabnike LEGO Mindstorms EV3  
[https://le-www-live-s.legocdn.com/sc/media/files/user-guides/ev3/ev3\\_user\\_guide\\_us-ab5d4fc71211c1edc77a7362bab2d88a.pdf?la=en-us](https://le-www-live-s.legocdn.com/sc/media/files/user-guides/ev3/ev3_user_guide_us-ab5d4fc71211c1edc77a7362bab2d88a.pdf?la=en-us)

Učni

pripomočki:

tabla, flomastri različnih barv, osebni računalnik, osnovni komplet LEGO Mindstorms EV3

Didaktična struktura ure:

1. Uvod (5 min) – uvodna motivacija
2. Ponovitev senzorjev (10 min) – osnovne lastnosti in uporaba senzorjev
3. Predstavitev vejitve (20 min) – blok Switch
4. Pogojevanje s senzorji (30 min)
5. Programiranje (20 min)
6. Diskusija (5 min)

Učitelj:	Dijaki:
<p>Na začetku ponovimo in osvežimo snov prejšnje ure.</p> <p>Ponovimo osnovne lastnosti senzorjev.</p> <p>Iz ponovitve senzorjev učno uro napeljemo na uporabo pogojev. Dijakom predstavimo programski blok Switch, ki omogoča uporabo pogojev. Z uporabo pogojev lahko določamo različne scenarije nadaljevanja programa.</p> <p>Dijake usmerjamo pri gradnji osnovnega programa z uporabo bloka Switch, ob tem pa razložimo različne funkcionalnosti tega bloka:</p> <ul style="list-style-type: none"> <li>• določanje pogoja</li> <li>• določanje možnosti</li> <li>• dodajanje/odvzemanje možnosti</li> </ul> <p>Postopoma zgradimo program, po katerem kocka glede na stanje pritisnega senzorja spusti različna zvoka.</p> <div data-bbox="215 985 933 1680" data-label="Diagram"> </div> <p>Dijake spodbudimo k razmišljanju o slabosti tega programa in jih usmerjamo k ugotovitvi, da mora tipka senzorja že biti pritisnjena, ko se program zažene, sicer se bo vedno izvedla druga opcija vejitve. Postopoma dodelamo program, tako da dodamo Wait blok, ki čaka na spremembo stanja pritisnega senzorja.</p>	<p>Poslušajo, sodelujejo, zapisujejo.</p> <p>Dijaki po naših usmeritvah gradijo program in sodelujejo.</p> <p>Dijaki po naših usmeritvah gradijo program in sodelujejo.</p>

Učitelj:	Dijaki:
<div data-bbox="268 324 1125 965" data-label="Diagram"> </div> <p data-bbox="237 1025 1129 1171">Zopet razmišljamo o slabostih in pridemo do ugotovitve, da se tokrat nikoli ne izvede druga možnost vejitve, ko tipka ni pritisnjena. Z dijaki razglabljamo o različnih možnostih dodelave programa.</p> <p data-bbox="237 1191 1142 1337">Dijakom damo eno ali več nalog, v katerih morajo uporabiti enega ali več senzorjev in pogojni blok. Dijake usmerjamo do pravilne rešitve in jih spodbujamo k ustvarjalnosti ter redoljubnemu sestavljanju programa.</p> <p data-bbox="237 1377 1098 1523">Primer naloge: Robot naj se pelje s konstanto hitrostjo, dokler ne pride na najmanjšo možno razdaljo do stene. Nato se obrne na desno in predvaja zvok "right".</p> <p data-bbox="237 1579 1008 1612">Naslednjo uro sledijo gnezdeni pogoji in večopravilnost.</p>	<p data-bbox="1201 555 1410 734">Dijaki po naših usmeritvah gradijo program in sodelujejo.</p> <p data-bbox="1201 1191 1404 1261">Dijaki rešujejo naloge.</p>

**UČNA PRIPRAVA**Predmet: **IP Informatika**

Program: gimnazija

Enota: **LEGO Mindstorms**Tema: **Gnezdeni pogoji in večopravilnost**

Tip učne enote: kombinirana

Zap. št.  
priprave:

7

Oddelek:

Datum:

Zap. št. ure:

13-14

Šola:

Učitelj:

Šolsko leto:

Operativni vzgojno-izobraževalni cilji:

Dijaki že poznajo pogoje, sedaj pa spoznajo še tehniko gnezdenja pogojev.  
Za pisanje kompleksnejših programov spoznajo večopravilnost.

Učne oblike: kombinirana

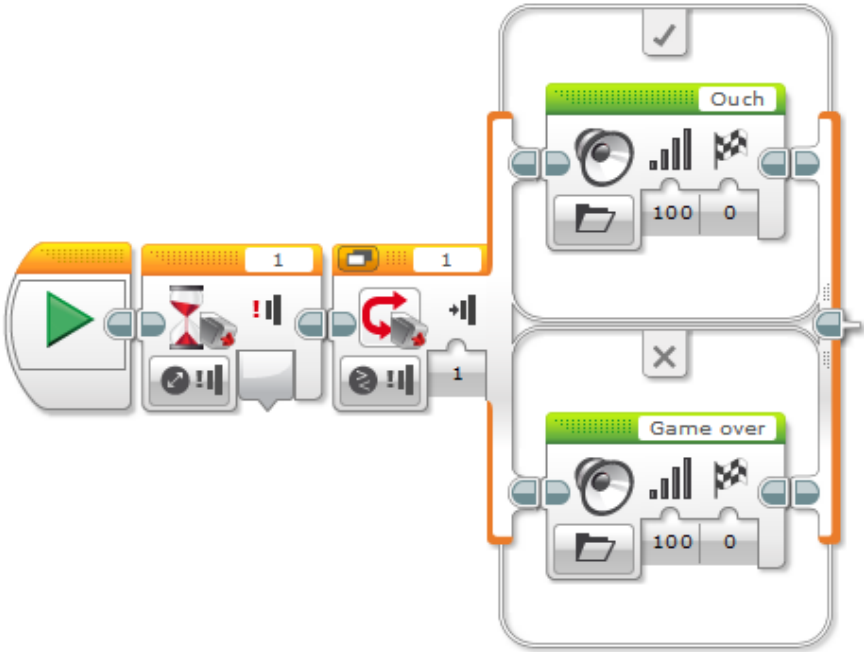
Učne metode: razlaga, diskusija, reševanje problemov

Viri: navodila za uporabnike LEGO Mindstorms EV3  
[https://le-www-live-s.legocdn.com/sc/media/files/user-guides/ev3/ev3\\_user\\_guide\\_us-ab5d4fc71211c1edc77a7362bab2d88a.pdf?la=en-us](https://le-www-live-s.legocdn.com/sc/media/files/user-guides/ev3/ev3_user_guide_us-ab5d4fc71211c1edc77a7362bab2d88a.pdf?la=en-us)

Učni  
pripomočki: tabla, flomastri različnih barv, osebni računalnik, osnovni komplet LEGO Mindstorms EV3

Didaktična struktura ure:

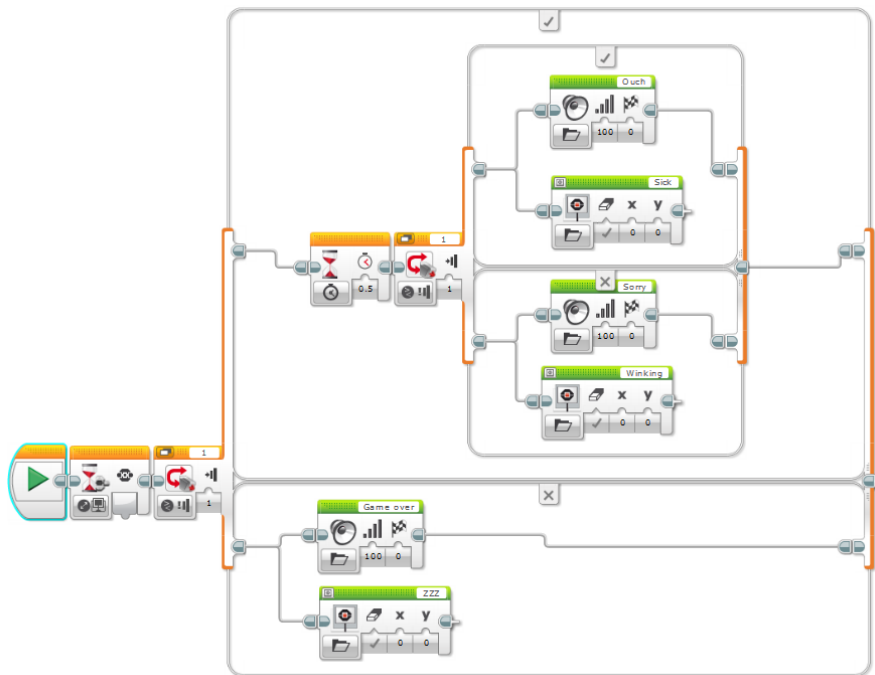
1. Uvod (5 min) – uvodna motivacija
2. Ponovitev vejitve (Switch) (10 min)
3. Predstavitev gnezdenja pogojev (20 min)
4. Predstavitev večopravilnosti – multitasking (20 min)
5. Programiranje (30 min)
6. Diskusija (5 min)

Učitelj:	Dijaki:
<p>Na začetku ponovimo in osvežimo snov prejšnje ure.</p> <p>Ponovimo uporabo bloka Switch in hitro zgradimo program, ki smo ga za primer zgradili na prejšnji uri.</p>  <p>Dijake spodbudimo k razmišljanju, kako bi zgradili program, ki bi ob pritisku tipke preveril, ali je tipka naknadno zadržana ali sproščena. Razmišljanje navežemo na gnezdenje pogojev in če je smiselno, lahko rešitev najprej podpremo z diagramom poteka na tabli. Postopoma dopolnimo program.</p>	<p>Poslušajo, sodelujejo, zapisujejo.</p> <p>Dijaki po naših usmeritvah gradijo program in sodelujejo.</p> <p>Dijaki po</p>

Učitelj:	Dijaki:
<div data-bbox="178 315 1090 1037" data-label="Diagram"> <p>The diagram shows a Scratch script for a game loop. It begins with a 'when green flag clicked' event block. This is followed by a 'say' block (with a speech bubble icon), a 'wait 0.5 seconds' block, and a 'repeat 1' loop block. Inside the loop, there is a 'say' block and a 'wait 1 second' block. After the loop, there is a 'Game over' block. The script is designed to run a sequence of actions and then repeat a specific part of it.</p> </div> <p data-bbox="164 1126 1165 1458">Na osnovi enakega programa (ali kakšnega drugega, npr. kakšen drug primer iz prejšnje ure) razmišljanje dijakov navežemo na situacije, ko bi od robota radi, da naenkrat naredi več stvari. Običajno je prvo ugibanje dijakov tako, da stvari nanizamo eno za drugo, zato najprej poskusimo to. Želimo namreč doseči, da se zvočni posnetki obogatijo z izrazi na ekranu. Z dijaki ugotovimo, da če bloke nizamo enega za drugim, se najprej izvede prvi in šele nato drugi, mi pa bi radi, da se izvedeta hkrati. Dijakom opišemo princip večopravnosti (multitasking) in ga prikažemo s praktičnim primerom.</p>	<p data-bbox="1193 293 1353 477">naših usmeritvah gradijo program in sodelujejo.</p> <p data-bbox="1193 1041 1353 1261">Dijaki po naših usmeritvah gradijo program in sodelujejo.</p>

Učitelj:

Dijaki:



Dijakom podamo eno ali več nalog, v katerih morajo uporabiti motorje, senzorje, pogoje, gnezdene pogoje in večopravilnost. Dijake usmerjamo do pravilne rešitve in jih spodbujamo k ustvarjalnosti ter redoljubnemu sestavljanju programa.

Primer naloge:

Robot naj se pelje s konstanto hitrostjo, dokler ne pride na najmanjšo možno razdaljo do stene. Ob vožnji naj spušča zvok motorja, ko se približa steni, pa naj spusti zvok zaustavitve. Nato naj čaka na pritisk tipke na kocki. Robot naj gre v smer, ki jo pritismo. Za vsako smer naj preveri, ali je pritisnjena še kakšna smer, in naj gre tudi po diagonalni.

Naslednjo uro sledi reševanje praktičnih nalog z do sedaj obravnavano snovjo.

Dijaki rešujejo naloge.

**UČNA PRIPRAVA**Predmet: **IP Informatika**

Program: gimnazija

Enota: **LEGO Mindstorms**Tema: **Gnezdeni pogoji in večopravilnost**

Tip učne enote: kombinirana

Zap. št.  
priprave:

8

Oddelek:

Datum:

Zap. št. ure:

15–16

Šola:

Učitelj:

Šolsko leto:

Operativni vzgojno-izobraževalni cilji:

Dijaki rešujejo izzive in utrjujejo do sedaj predelano snov.

Učne oblike: kombinirana

Učne metode: razlaga, diskusija, reševanje problemov

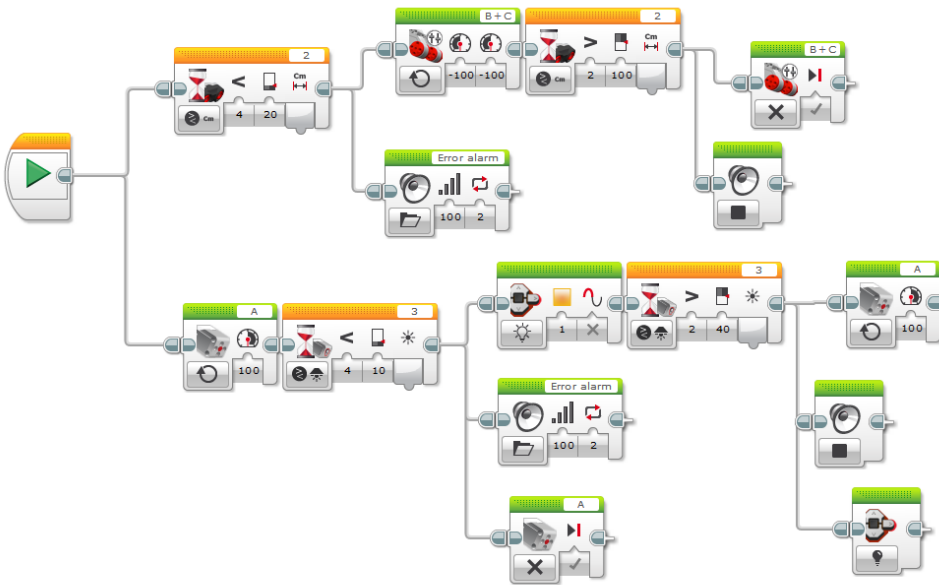
Viri: navodila za uporabnike LEGO Mindstorms EV3  
[https://le-www-live-s.legocdn.com/sc/media/files/user-guides/ev3/ev3\\_user\\_guide\\_us-ab5d4fc71211c1edc77a7362bab2d88a.pdf?la=en-us](https://le-www-live-s.legocdn.com/sc/media/files/user-guides/ev3/ev3_user_guide_us-ab5d4fc71211c1edc77a7362bab2d88a.pdf?la=en-us)

Učni  
pripomočki: tabla, flomastri različnih barv, osebni računalnik, osnovni komplet  
LEGO Mindstorms EV3

Didaktična struktura ure:

1. Uvod (5 min) – uvodna motivacija
2. Reševanje izzivov (70 min)
3. Diskusija (15 min)



Učitelj:	Dijaki:
<p>Na začetku ponovimo in osvežimo snov prejšnje ure.</p> <p>Dijakom podamo enega ali več izzivov, ki so predvideni za reševanje. Dijake med reševanjem usmerjamo do pravilne rešitve in jih spodbujamo k ustvarjalnosti ter redoljubnemu sestavljanju programa.</p> <p>Primer izziva: Mobilna državna varnostna utrdbo  Zgradite mobilno državno varnostno utrdbo. Ta naj ima dvignjeno zastavo, ki se ves čas vrsti v smeri urinega kazalca, dokler se ne stemni. Ko se stemni, naj se prižgejo vse varnostne luči in zaženejo zvočni efekti za odganjanje sovražnikov. Ko se zdani, naj se spet vrti zastava. Utrdba naj ves čas budno spremlja, ali se bliža sovražnik. Če se sovražnik približa na nevarno razdaljo, naj utrdbo sproži alarm in z vso hitrostjo odpelje vzvratno, ob tem pa oddaja opozorilne zvoke, dokler ni spet na varni razdalji pred sovražnikom.</p>  <p>Na koncu skupaj z dijaki pregledamo rešitve in razpravljamo o boljših in slabših načinih.</p> <p>Naslednjo uro sledijo zanke.</p>	<p>Rešujejo izziv/e v skupinah.</p>

**UČNA PRIPRAVA**Predmet: **IP Informatika**

Program: gimnazija

Enota: **LEGO Mindstorms**Tema: **Zanke in pogoji za izstop**

Tip učne enote: kombinirana

Zap. št.  
priprave:

9

Oddelek:

Datum:

Zap. št. ure:

17–18

Šola:

Učitelj:

Šolsko leto:

Operativni vzgojno-izobraževalni cilji:

Dijaki spoznajo zanke in pogoje za izstop.

Učne oblike: kombinirana

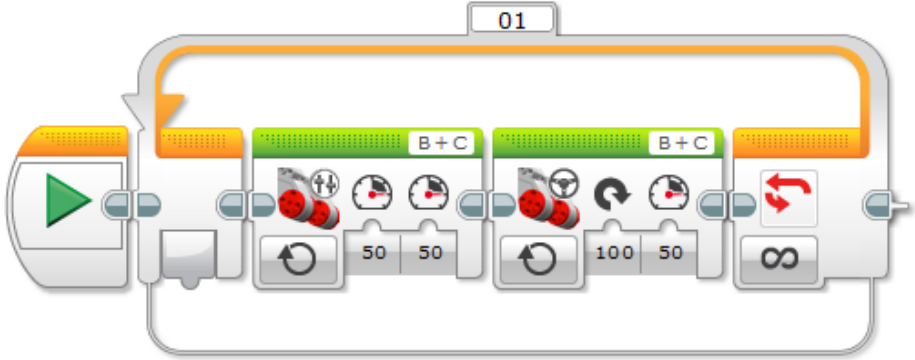
Učne metode: razlaga, diskusija, reševanje problemov

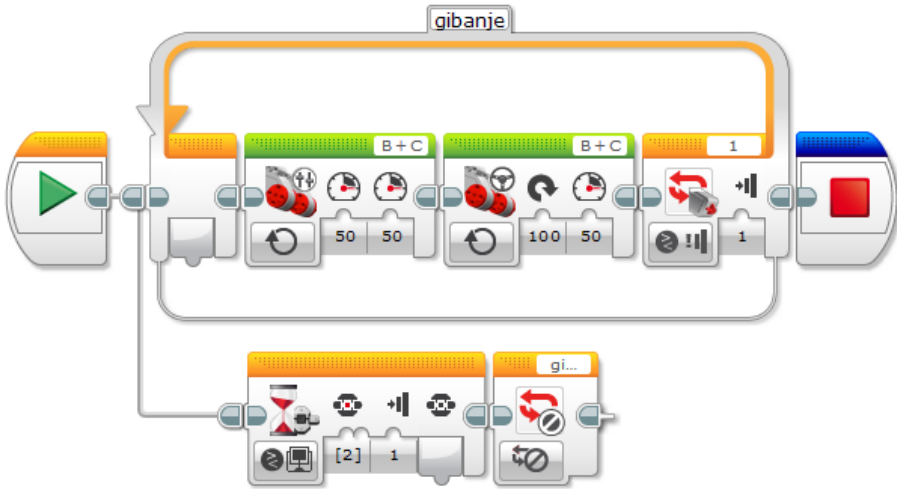
Viri: navodila za uporabnike LEGO Mindstorms EV3  
[https://le-www-live-s.legocdn.com/sc/media/files/user-guides/ev3/ev3\\_user\\_guide\\_us-ab5d4fc71211c1edc77a7362bab2d88a.pdf?la=en-us](https://le-www-live-s.legocdn.com/sc/media/files/user-guides/ev3/ev3_user_guide_us-ab5d4fc71211c1edc77a7362bab2d88a.pdf?la=en-us)

Učni  
pripomočki: tabla, flomastri različnih barv, osebni računalnik, osnovni komplet  
LEGO Mindstorms EV3

Didaktična struktura ure:

1. Uvod (5 min) – uvodna motivacija
2. Ponovitev zank z diagramom poteka (5 min)
3. Predstavitev zanke v EV3 okolju (30 min)
4. Prekinitev zanke (20 min)
5. Reševanje problemov (25 min)
6. Diskusija (5 min)

Učitelj:	Dijaki:
<p>Na začetku ponovimo in osvežimo snov prejšnje ure.</p> <p>Dijake spodbudimo k razmišljanju o zankah s primerom iz vsakdanjega življenja (ponavljanje opravil).</p> <p>Če je smiselno, lahko za uvod sestavimo diagram poteka na tablo, ki uporablja zanko.</p> <p>Skupaj z dijaki zgradimo prvo enostavno neskončno zanko.</p>  <p>Dijakom ob sami gradnji zanke predstavimo osnovne dele bloka zanke:</p> <ul style="list-style-type: none"> <li>• telo zanke,</li> <li>• pogoj,</li> <li>• števec, ki ga za sedaj še ne znamo uporabiti,</li> <li>• ime zanke.</li> </ul> <p>Dijake spodbudimo k razmišljanju o tem, kako bi zanko ustavili. Skupaj raziščemo različne pogoje; običajno je prvi predlog dijakov eden izmed najenostavnejših pogojev:</p> <ul style="list-style-type: none"> <li>• Count (števec) ali</li> <li>• Time Indicator (časovno).</li> </ul> <p>Skupaj z dijaki preizkusimo enega in drugega ter opazujemo. Dijakom predstavimo, kako lahko v programski opremi EV3 spremljajo, kateri blok se trenutno izvaja. Skupaj z dijaki opazujemo, kaj pomeni števec in kolikokrat se zanka v tem načinu izvede.</p> <p>V naslednjem koraku se osredotočimo na ostale pogoje za izstop iz zanke, ki so na voljo – senzorji. Dijake napotimo, da na svojega robota vgradijo Touch sensor. Nato na naši vnaprej zgrajeni zanki spremenimo pogoj, tako da uporabimo Touch sensor. Dijake</p>	<p>Poslušajo, sodelujejo, zapisujejo.</p> <p>Sodelujejo, programirajo .</p> <p>Sodelujejo, zapisujejo, raziskujejo.</p> <p>Sodelujejo, preizkušajo.</p> <p>Sodelujejo, preizkušajo.</p>

Učitelj:	Dijaki:
<p>spodbudimo, da preizkusijo različne možnosti.</p> <p>V nadaljevanju dijake spodbudimo k razmišljanju o tem, kaj vse lahko prekine zanko. Prvi odgovor je običajno enostaven: pogoj. Dijakom postavimo manjši izziv. Pogoj zanke, ki so jo že sestavili, naj ostane enak (glede na senzor), znotraj zanke pa naj v vsaki ponovitvi zanke preverjajo, ali je morda pritisnjena sredinska tipka na kocki in v tem primeru izstopijo iz zanke.</p> <p>Dijakom pustimo nekaj minut za samostojno raziskovanje, nato pa skupaj vodeno pogledamo možne rešitve.</p> <p>Z zgoraj omenjenim primerom predstavimo blok za prekinitev zanke. Dijakom predstavimo pojem imena zanke in zakaj je potreben – brez imena zanke bloku za prekinitev ne moremo povedati, katero zanko naj ta blok prekine. Dijake spodbudimo, naj si za ime zanke izberejo nedvomne pojme, ki dobro opišejo, kaj zanka dela. Razložimo, da je to potrebno in priporočeno zaradi obsežnejših programov z velikim številom zank. Če je ime dobro, zelo opisno, lahko hitro vemo, katero zanko moramo v določenem trenutku prekiniti.</p> <p>Problem je mogoče rešiti na več načinov. Dijake usmerjamo k rešitvi problema s stavkom Switch ter Wait in z večopravilnostjo. Na sliki je ena izmed možnih rešitev z uporabo večopravilnosti in bloka Wait.</p>  <p>Dijakom ponudimo krajši izziv.</p> <p>Primer izziva: Robot naj se pelje do ovire in vsakič tik pred oviro zavije desno. Hkrati naj zavoj označi še z zvočnimi, s svetlobnimi in slikovnimi efekti. Če</p>	<p>Sodelujejo, razmišljajo, preizkušajo.</p> <p>Sodelujejo, preizkušajo.</p> <p>Sodelujejo, diskutirajo o različnih možnostih.</p>
<p>Razmišljajo, raziskujejo, programirajo ,</p>	<p>Razmišljajo, raziskujejo, programirajo ,</p>

Učitelj:	Dijaki:
kadarkoli ugasnemo luč, naj se robot ustavi. Razmislite, kaj se zgodi, če se robot znajde v kotu – ko zavije v desno, takoj naleti na novo oviro.	rešujejo problem.
Naslednjo uro sledijo spremenljivke in povezovanje podatkov.	

**UČNA PRIPRAVA**Predmet: **IP Informatika**

Program: gimnazija

Enota: **LEGO Mindstorms**Tema: **Spremenljivke, povezovanje podatkov, matematične operacije**

Tip učne enote: kombinirana

Zap. št.  
priprave:

10

Oddelek:

Datum:

Zap. št. ure:

19–20

Šola:

Učitelj:

Šolsko leto:

Operativni vzgojno-izobraževalni cilji:

Dijaki spoznajo spremenljivke in delo z njimi ter tehniko povezovanja podatkov.

Dijaki spoznajo matematične in logične operacije v programski opremi EV3.

Učne oblike: kombinirana

Učne metode: razlaga, diskusija, reševanje problemov

Viri:

navodila za uporabnike LEGO Mindstorms EV3  
[https://le-www-live-s.legocdn.com/sc/media/files/user-guides/ev3/ev3\\_user\\_guide\\_us-ab5d4fc71211c1edc77a7362bab2d88a.pdf?la=en-us](https://le-www-live-s.legocdn.com/sc/media/files/user-guides/ev3/ev3_user_guide_us-ab5d4fc71211c1edc77a7362bab2d88a.pdf?la=en-us)

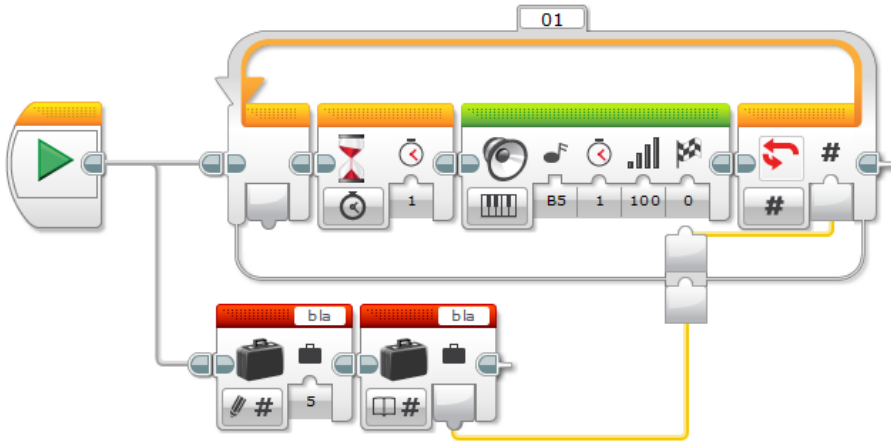
Učni

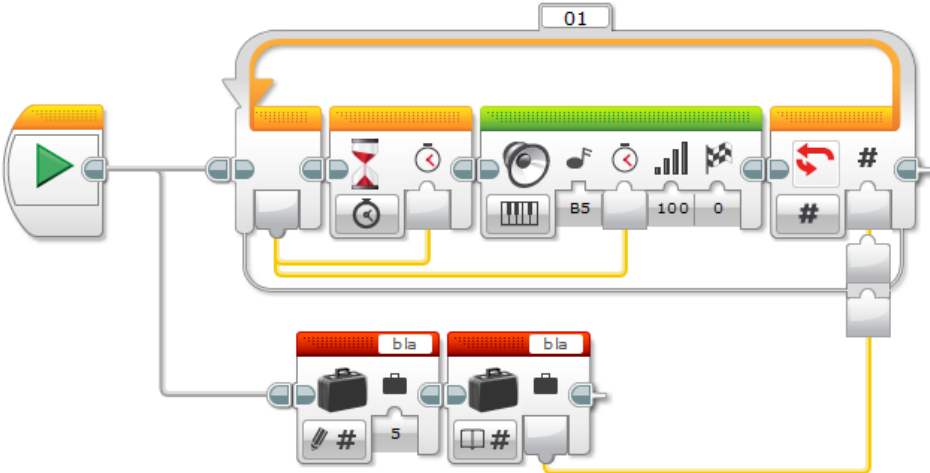
pripomočki:

tabla, flomastri različnih barv, osebni računalnik, osnovni komplet LEGO Mindstorms EV3

Didaktična struktura ure:

1. Uvod (5 min) – uvodna motivacija
2. Ponovitev pojma spremenljivke z diagramom poteka (10 min)
3. Predstavitev spremenljivke v EV3 okolju (10 min)
4. Predstavitev tehnike povezovanja podatkov (10 min)
5. Predstavitev matematičnih in logičnih operacij v okolju EV3 (30 min)
6. Reševanje problemov (20 min)
7. Diskusija (5 min)

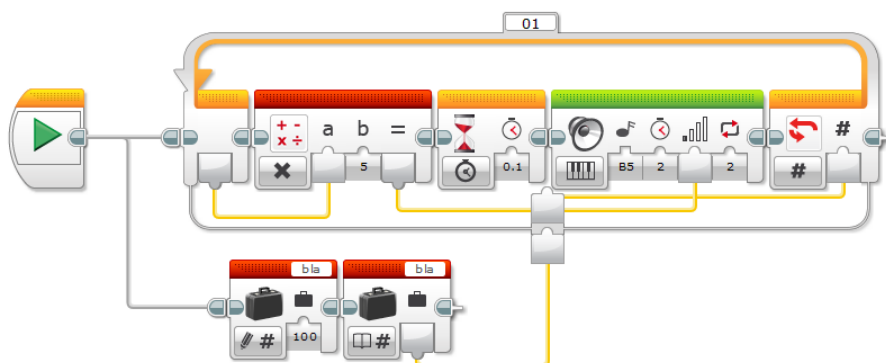
Učitelj:	Dijaki:
<p>Na začetku ponovimo in osvežimo snov prejšnje ure.</p> <p>Dijake spodbudimo k razmišljanju o spremenljivkah s primerom iz vsakdanjega življenja (ponavljanje opravil in štetje s števcem).</p> <p>Če je smiselno, lahko za uvod sestavimo diagram poteka na tablo, ki uporablja spremenljivke.</p> <p>Z dijaki razmišljamo, kdaj je smiselno, da neko stvar »zapakiramo« v spremenljivko – kadar obstaja možnost, da bomo podatek potrebovali večkrat ali na več mestih.</p> <p>Skupaj z dijaki zgradimo prvi program z uporabo spremenljivke. Dijakom predstavimo blok spremenljivke in dva različna načina dela s spremenljivko (read/write).</p> <p>Podobno kot smo to storili pri zankah, tudi pri spremenljivkah dijake spodbudimo, da za imena spremenljivk izbirajo pojme, ki so opisni.</p> <p>Najprej postavimo blok spremenljivke in v njo spravimo numerični podatek (npr. 5). Nato postavimo še zanko in za števec zanke uporabimo vrednost spremenljivke.</p> 	<p>Poslušajo, sodelujejo, zapisujejo.</p> <p>Sodelujejo, programirajo .</p>
<p>Na ta način smo že uporabili tehniko povezovanja podatkov, ki jo sedaj lahko predstavimo dijakom. Pojasnimo jim razliko med vstopnimi in izstopnimi podatki ter kako jih v programski opremi EV3 ločimo (ušesek kvadratika obrnjen navzgor/navzdol). Na primeru jim pokažemo, da ima blok spremenljivke v načinu »write« vstopni argument, v načinu »read« pa izstopni argument.</p> <p>Dijake v nadaljevanju spodbudimo, da preizkusijo, kaj se zgodi, če v prejšnjem programu števec zanke, ki se nahaja na levi strani bloka</p>	<p>Sodelujejo, zapisujejo, raziskujejo.</p> <p>Sodelujejo, preizkušajo,</p>

Učitelj:	Dijaki:
<p>zanke, povežejo z vstopnim argumentom bloka Wait.</p> <p>Dijaki ugotovijo, da se program v vsaki ponovitvi zanke dlje zakasni. Na ta način sami ugotovijo, kako in kje bi lahko uporabili števec zanke in povezovanje podatkov. Spodbudimo jih, da števec zanke povežejo še z argumentom trajanja zvoka. Na ta način jim pokažemo, da izstopne podatke lahko povezujemo tudi na več vstopnih podatkov.</p>  <p>V nadaljevanju dijake spodbudimo, da števec zanke povežejo z vstopnim podatkom za glasnost zvoka. Dijaki ugotovijo, da se v delovanju programa glasnost praktično ne spremeni. Nato jih napotimo, da števec zanke povečajo na 100 in spremljajo, kaj se dogaja. Spodbudimo jih k razmišljanju, kako bi, še vedno z uporabo števca, pospešili oziroma povečali stopnjo povečevanja glasnosti. Skupaj pridemo do rešitve, da bi lahko števec vsakič pomnožili s konstanto (npr. 5).</p> <p>Na ta način v program vpeljemo blok za matematične operacije (Math). Vstopni argument enega faktorja matematičnega bloka povežemo s števcem zanke, produkt pa povežemo z glasnostjo. Dijake spodbudimo, da poskusijo spreminjati vrednosti in spremljajo delovanje.</p>	<p>opazujejo delovanje.</p> <p>Sodelujejo, preizkušajo.</p> <p>Sodelujejo, razmišljajo, preizkušajo.</p> <p>Sodelujejo, razmišljajo, preizkušajo.</p>



Učitelj:

Dijaki:



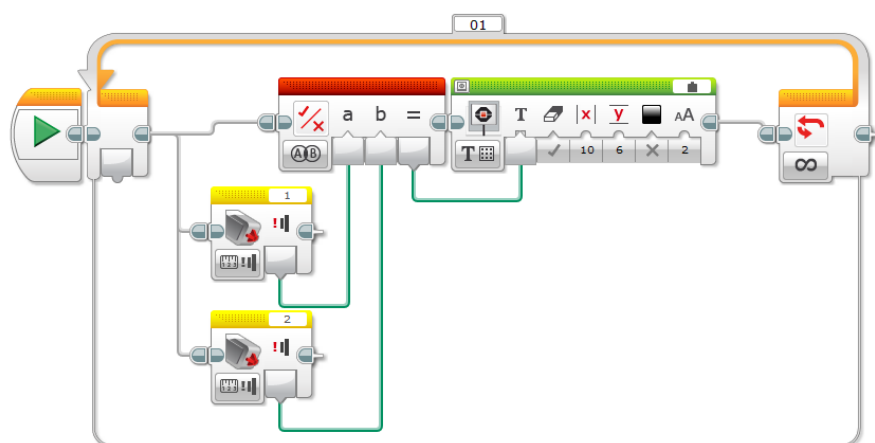
Dijake spodbudimo, da preizkusijo še ostale matematične operacije. Predstavimo jim možnost Advanced in jim pokažemo, kako lahko zapišejo svojo funkcijo za izračun ali pa uporabijo še druge funkcije. Dijakom predstavimo še bloke Round, Compare in Random ter jih spodbudimo k preizkušanju teh blokov.

Nato preidemo na logične funkcije.

Za dijake vnaprej pripravimo program, ki si ga dijaki prenesejo na računalnik.

Program je narejen tako, da uporablja dva pritiska senzorja in na ekranu izpisuje rezultat logične operacije izhodnih podatkov obeh senzorjev.

Program:



Dijake spodbudimo k premisleku, kako program deluje in kaj prikazuje na zaslonu.

Dijakom predstavimo blok za logične operacije. Vodenno preverimo različne funkcije, ki so na voljo. Skupaj z dijaki preverimo delovanje

Razmišljajo, raziskujejo, programirajo, rešujejo problem.

Prenesejo program, sodelujejo

Zaženejo program, raziskujejo delovanje, opazujejo.

Učitelj:	Dijaki:																																																							
<p>različnih funkcij in na tabli sestavimo tabele funkcij. Obrazložimo pojem boolean tipa podatka, ki pozna samo dve vrednosti: 1/0 (true/false).</p> <p><b>AND</b> (in – konjunkcija) – 1, ko sta ob vhoda enaka 1</p> <table><tr><th>Vhod 1</th><th>Vhod 2</th><th>Rezultat</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table> <p><b>OR</b> (ali – disjunkcija) – 1, ko je vsaj eden od vhodov enak 1</p> <table><tr><th>Vhod 1</th><th>Vhod 2</th><th>Rezultat</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table> <p><b>XOR</b> (ekskluzivni ali) – 1, ko je natančno en vhod enak 1</p> <table><tr><th>Vhod 1</th><th>Vhod 2</th><th>Rezultat</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>1</td><td>0</td></tr></table> <p><b>NOT</b> (ne – negacija) - laž</p> <table><tr><th>Vhod 1</th><th>Rezultat</th></tr><tr><td>0</td><td>1</td></tr><tr><td>1</td><td>0</td></tr><tr><td>0</td><td>1</td></tr><tr><td>1</td><td>0</td></tr></table> <p>Naslednjo uro sledi nov programski izziv.</p>	Vhod 1	Vhod 2	Rezultat	0	0	0	1	0	0	0	1	0	1	1	1	Vhod 1	Vhod 2	Rezultat	0	0	0	1	0	1	0	1	1	1	1	1	Vhod 1	Vhod 2	Rezultat	0	0	0	1	0	1	0	1	1	1	1	0	Vhod 1	Rezultat	0	1	1	0	0	1	1	0	<p>Poslušajo, sodelujejo, zapisujejo, raziskujejo.</p>
Vhod 1	Vhod 2	Rezultat																																																						
0	0	0																																																						
1	0	0																																																						
0	1	0																																																						
1	1	1																																																						
Vhod 1	Vhod 2	Rezultat																																																						
0	0	0																																																						
1	0	1																																																						
0	1	1																																																						
1	1	1																																																						
Vhod 1	Vhod 2	Rezultat																																																						
0	0	0																																																						
1	0	1																																																						
0	1	1																																																						
1	1	0																																																						
Vhod 1	Rezultat																																																							
0	1																																																							
1	0																																																							
0	1																																																							
1	0																																																							

**UČNA PRIPRAVA**Predmet: **IP Informatika**

Program: gimnazija

Enota: **LEGO Mindstorms**Tema: **Spremenljivke, povezovanje podatkov, matematične operacije**

Tip učne enote: kombinirana

Zap. št.  
priprave:

11

Oddelek:

Datum:

Zap. št. ure:

21–22

Šola:

Učitelj:

Šolsko leto:

Operativni vzgojno-izobraževalni cilji:

Dijaki rešujejo programski izziv, v sklopu katerega morajo uporabljati zanke, pogoje, večopravilnost, matematične in logične operacije.

Učne oblike: kombinirana

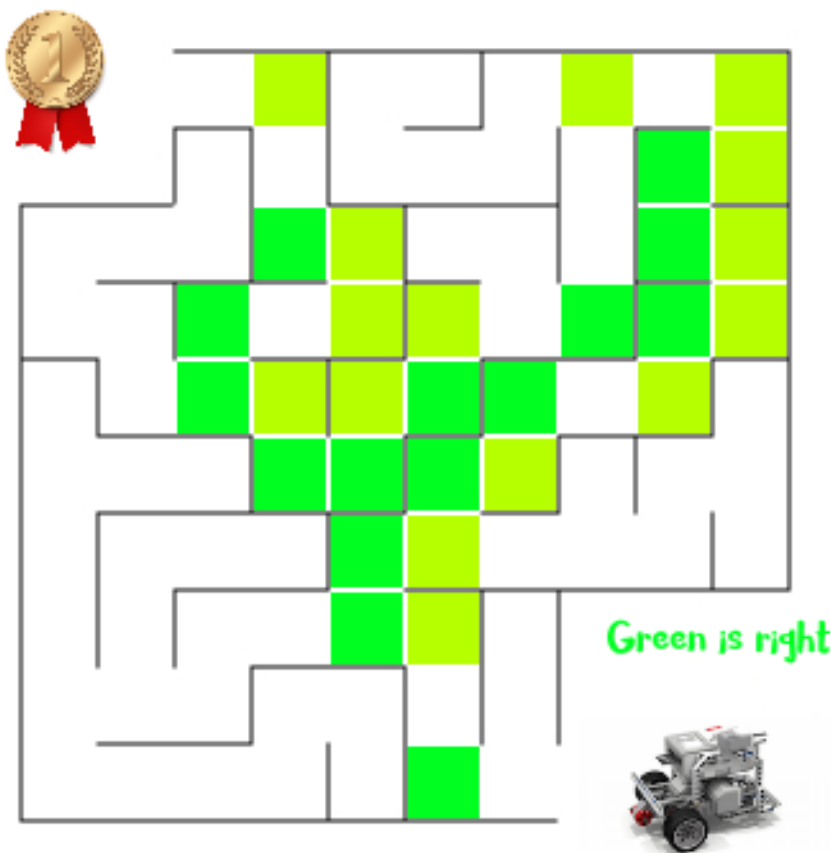
Učne metode: razlaga, diskusija, reševanje problemov

Viri: navodila za uporabnike LEGO Mindstorms EV3  
[https://le-www-live-s.legocdn.com/sc/media/files/user-guides/ev3/ev3\\_user\\_guide\\_us-ab5d4fc71211c1edc77a7362bab2d88a.pdf?la=en-us](https://le-www-live-s.legocdn.com/sc/media/files/user-guides/ev3/ev3_user_guide_us-ab5d4fc71211c1edc77a7362bab2d88a.pdf?la=en-us)

Učni  
pripomočki: tabla, flomastri različnih barv, osebni računalnik, osnovni komplet LEGO Mindstorms EV3

Didaktična struktura ure:

1. Uvod (5 min) – uvodna motivacija
2. Navodila za izziv in namigi (10 min)
3. Reševanje izziva (60 min)
4. Diskusija (15 min)

Učitelj:	Dijaki:
<p>Na začetku ponovimo in osvežimo snov prejšnje ure.</p> <p>Dijakom predstavimo izziv, ki naj ga rešujejo samostojno.</p> <p>Primer izziva:          Za dijake pripravimo labirint, po katerem bodo peljali robota. Najti morajo pravilen način, da bo robot sam našel pravilno pot na drugo stran labirinta.</p> <div data-bbox="252 649 1086 1498">  </div> <p>Navodila:          Znašel si se pred vrati labirinta. Po nekaj urah razmišljanja in poskušanja naključnih smeri si se vrnil na začetek in opazil vzorec. Na nekaterih križiščih so na tleh obarvane plošče (rumene ali zelene), na vratih labirinta pa si opazil napis: »Zelena je pravilna« (ang. Green is right). Kako boš prišel na drugo stran labirinta, kjer te čaka ocena 5 pri informatiki?</p> <p>Za napredne lahko podamo še dodatne izzive:</p> <ul style="list-style-type: none"> <li>- poskrbi, da bo robot ves čas držal pravo smer, opravljal ravno pravšnje obrate itd.</li> </ul>	<p>Čim bolj samostojno rešujejo nalogo, razmišljajo, raziskujejo.</p>

Učitelj:	Dijaki:
<ul style="list-style-type: none"><li>- če robot zaporedoma zavije levo in nato desno, naj spusti poseben zvočni signal</li></ul> <p>Dijake usmerjamo do pravilne rešitve in jih spodbujamo k ustvarjalnosti ter redoljubnemu sestavljanju programa.</p> <p>Naslednjo uro sledi nov programski izziv.</p>	

**UČNA PRIPRAVA**Predmet: **IP Informatika**

Program: gimnazija

Enota: **LEGO Mindstorms**Tema: **Seznami in indeksiranje**

Tip učne enote: kombinirana

Zap. št.  
priprave:

12

Oddelek:

Datum:

Zap. št. ure:

23–24

Šola:

Učitelj:

Šolsko leto:

Operativni vzgojno-izobraževalni cilji:

Dijaki že poznajo spremenljivke.

Dijaki spoznajo sezname in pojem indeksiranje.

Učne oblike: kombinirana

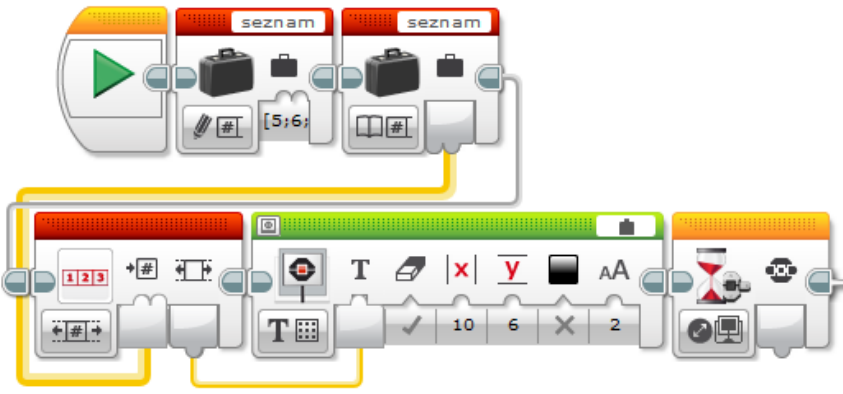
Učne metode: razlaga, diskusija, reševanje problemov

Viri: navodila za uporabnike LEGO Mindstorms EV3  
[https://le-www-live-s.legocdn.com/sc/media/files/user-guides/ev3/ev3\\_user\\_guide\\_us-ab5d4fc71211c1edc77a7362bab2d88a.pdf?la=en-us](https://le-www-live-s.legocdn.com/sc/media/files/user-guides/ev3/ev3_user_guide_us-ab5d4fc71211c1edc77a7362bab2d88a.pdf?la=en-us)

Učni  
pripomočki: tabla, flomastri različnih barv, osebni računalnik, osnovni komplet  
LEGO Mindstorms EV3, A4 listi z naključnimi besedami

Didaktična struktura ure:

1. Uvod (5 min) – uvodna motivacija
2. Ponovitev spremenljivk (5 min)
3. Predstavitev seznama z diagramom poteka (15 min)
4. Predstavitev blokov za seznam v programski opremi EV3 (30 min)
5. Reševanje problemov (30 min)
6. Diskusija (5 min)

Učitelj:	Dijaki:
<p>Na začetku ponovimo in osvežimo snov prejšnje ure.</p> <p>Z dijaki ponovimo koncept spremenljivk in dela z njimi.</p> <p>Z enostavnim diagramom poteka obnovimo koncept seznama (spremenljivka, ki vsebuje več vrednosti) in pojem indeksiranja. Ponovimo, da so v računalniškem svetu indeksi običajno razvrščeni od 0 dalje.</p> <p>Pripravljene imamo A4 liste z naključnimi besedami (npr. živali, rastline). Prosimo nekaj dijakov, da pridejo pred tablo, in vsakemu damo po en list s številko. Ostale dijake spodbudimo, da odgovarjajo na naša vprašanja:</p> <ul style="list-style-type: none"> <li>• Kakšna je dolžina našega seznama?</li> <li>• Kakšen je indeks prvega elementa?</li> <li>• Kakšen je indeks elementa X?</li> <li>• Kateri element se nahaja na indeksu X?</li> <li>• Če odstranimo element X, kakšna je potem dolžina seznama?</li> <li>• Kaj se je zgodilo z indeksi?</li> </ul> <p>Dijakom predstavimo bloka, ki sta primerna za delo s seznamami:</p> <ul style="list-style-type: none"> <li>• variable in</li> <li>• array operations</li> </ul> <p>Z dijaki sestavimo enostaven seznam števil. Opazimo, da programska oprema EV3 podpira zgolj sezname števil in sezname Booleanov. Spoznamo blok za ustvarjanje seznamov in blok za branje seznama. Zgradimo enostaven program, ki na ekran kocke izpiše dolžino seznama.</p>  <p>Opazimo, da sta kvadrata za vhodne in izhodne podatke pri seznamu drugačna kot pri drugih spremenljivkah. Prav tako je drugačna »žica« za povezovanje podatkov.</p>	<p>Poslušajo, sodelujejo, zapisujejo.</p> <p>Poslušajo, sodelujejo, zapisujejo.</p> <p>Sodelujejo, programirajo.</p>

Učitelj:	Dijaki:
<p>Dijake spodbudimo k spreminjanju seznama in spremljanju, kako se spreminja vrednost, ki jo kocka izpisuje.</p> <p>Z dijaki preverimo različne opcije bloka Array Operations:</p> <ul style="list-style-type: none"><li>• append,</li><li>• read at index,</li><li>• write at index,</li><li>• length.</li></ul> <p>Operacije skušamo razložiti s številkami iz listov, ki jih držijo dijaki.</p> <p>V nadaljevanju dijakom postavimo izziv: Poskusite narediti program, ki bo v seznam zapisoval števec zanke. Na koncu naj se na ekranu izpiše seštevek vseh števil v seznamu.</p> <p>Dijakom najprej pustimo nekaj časa, da poskusijo problem razumeti sami in se začnejo ukvarjati z njim. Nato jih vodimo do pravilne rešitve.</p> <p>Problem poskusimo razdeliti na več manjših problemov:</p> <ul style="list-style-type: none"><li>• Kako bomo števec zanke zapisovali v seznam?</li><li>• Kako bomo izračunali seštevek vrednosti seznama?</li></ul> <p>Naslednjo uro sledijo vaje s senzorji in kalibracijo.</p>	<p>Sodelujejo, preizkušajo, opazujejo delovanje, poskušajo rešiti problem.</p>



**UČNA PRIPRAVA**Predmet: **IP Informatika**

Program: gimnazija

Enota: **LEGO Mindstorms**Tema: **Senzorji, spremljanje vrednosti in kalibracija**

Tip učne enote: kombinirana

Zap. št.  
priprave:

13

Oddelek:

Datum:

Zap. št. ure:

25–26

Šola:

Učitelj:

Šolsko leto:

Operativni vzgojno-izobraževalni cilji:

Dijaki že poznajo senzorje in njihovo osnovno delovanje. Senzorje preizkusijo in bolj podrobno preučijo vrednosti, ki jih senzorji vračajo. Dijaki osvojijo kalibracijo senzorjev.

Učne oblike: kombinirana

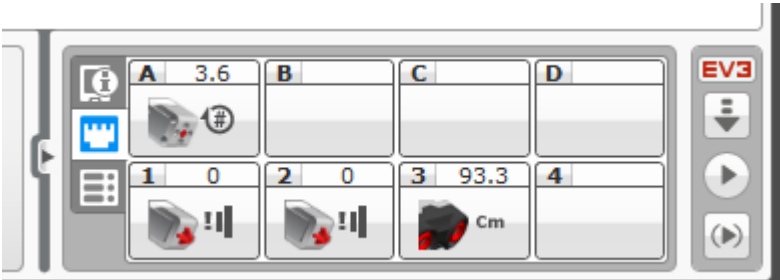
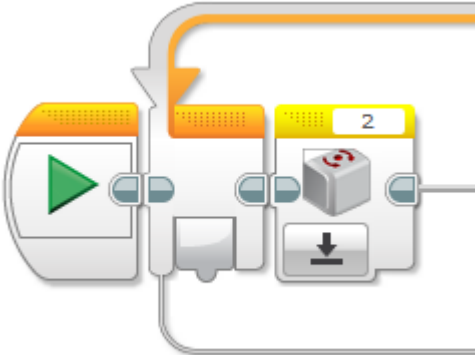
Učne metode: razlaga, diskusija, reševanje problemov

Viri: navodila za uporabnike LEGO Mindstorms EV3  
[https://le-www-live-s.legocdn.com/sc/media/files/user-guides/ev3/ev3\\_user\\_guide\\_us-ab5d4fc71211c1edc77a7362bab2d88a.pdf?la=en-us](https://le-www-live-s.legocdn.com/sc/media/files/user-guides/ev3/ev3_user_guide_us-ab5d4fc71211c1edc77a7362bab2d88a.pdf?la=en-us)

Učni  
pripomočki: tabla, flomastri različnih barv, osebni računalnik, osnovni komplet LEGO Mindstorms EV3

Didaktična struktura ure:

1. Uvod (5 min) – uvodna motivacija
2. Ponovitev osnov senzorjev (15 min)
3. Predstavitev spremljanja vrednosti senzorjev (15 min)
4. Kalibracija žiro senzorja (10 min)
5. Kalibracija barvnega senzorja (10 min)
6. Reševanje problemov (30 min)
7. Diskusija (5 min)

Učitelj:	Dijaki:
<p>Na začetku ponovimo in osvežimo snov prejšnje ure.</p> <p>Z dijaki ponovimo koncepte različnih senzorjev in dela z njimi.</p> <p>Dijakom predstavimo način za spremljanje različnih vrednosti, ki nam jih senzorji vračajo, in jih spodbudimo, da izdelajo programe, ki shranijo vrednosti senzorjev v spremenljivke in jih nato prikazujejo na ekranu kocke.</p>  <p>Predstavimo kalibracijo žiro senzorja.</p> <p>Dijakom predstavimo izziv:</p> <p>Na robot vgradite žiro senzor. Najprej naj se robot pelje naprej in naredi nekaj naključnih zavojev (lahko poskusite uporabiti blok Random), nato pa naj naredi natančen 90° zavoj.</p> <p>Izziv lahko z dijaki poskusimo rešiti skupaj, vodeno. V pogoj zanke nastavimo vrednost žiro senzorja, ki naj bo večja ali enaka 90°.</p> <p>Dijaki ugotovijo, da program ne deluje pravilno. Spodbudimo jih, da ugotovijo, zakaj. Predstavimo način kalibracije z uporabo možnosti Reset.</p> 	<p>Poslušajo, sodelujejo, zapisujejo.</p> <p>Sodelujejo, programirajo.</p> <p>Sodelujejo, programirajo.</p>

Učitelj:

Dijaki:

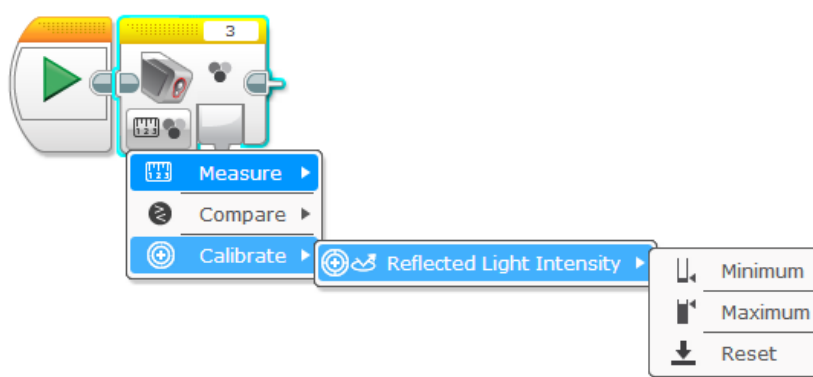
Predstavimo kalibracijo barvnega senzorja, ki je drugačna od kalibracije žiro senzorja.

Dijakom zopet postavimo izziv:

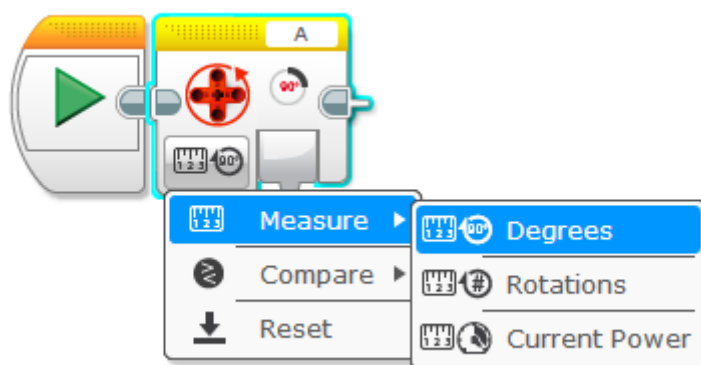
Izdelajte budilko. Ko postane dovolj svetlo, naj robot začne spuščati zvok. Ko dijaki končajo s programiranjem in preizkusijo program, čim bolj zatemnimo prostor (a ne do popolne teme). Ko dijaki še enkrat preizkusijo program, ugotovijo, da ta ne deluje več pravilno.

Skupaj z dijaki poskusimo program popraviti z uporabo ene od opcij za kalibracijo barvnega senzorja (minimum/maximum, reset).

Sodelujejo, programirajo.



Predstavimo način, kako motor spremenimo v senzor obratov ali moči.



Naslednjo uro sledi razhroščevanje.

**UČNA PRIPRAVA**Predmet: **IP Informatika**

Program: gimnazija

Enota: **LEGO Mindstorms**Tema: **Razhroščevanje (debugging)**

Tip učne enote: kombinirana

Zap. št.  
priprave:

14

Oddelek:

Datum:

Zap. št. ure:

27–28

Šola:

Učitelj:

Šolsko leto:

Operativni vzgojno-izobraževalni cilji:

Dijaki spoznajo pojem in tehnike razhroščevanja.

Dijaki spoznajo, kako izvajamo debugging v programski opremi EV3.

Učne oblike: kombinirana

Učne metode: razlaga, diskusija, reševanje problemov

Viri:

navodila za uporabnike LEGO Mindstorms EV3

[https://le-www-live-s.legocdn.com/sc/media/files/user-guides/ev3/ev3\\_user\\_guide\\_us-ab5d4fc71211c1edc77a7362bab2d88a.pdf?la=en-us](https://le-www-live-s.legocdn.com/sc/media/files/user-guides/ev3/ev3_user_guide_us-ab5d4fc71211c1edc77a7362bab2d88a.pdf?la=en-us)

Učni

pripomočki:

tabla, flomastri različnih barv, osebni računalnik, osnovni komplet LEGO Mindstorms EV3

Didaktična struktura ure:

1. Uvod (5 min) – uvodna motivacija
2. Predstavitev pojma razhroščevanja (15 min)
3. Predstavitev tehnik razhroščevanja (15 min)
4. Primer razhroščevanja z diagramom poteka (20 min)
5. Primer razhroščevanja s programsko opremo EV3 (30 min)
6. Diskusija (5 min)

Učitelj:	Dijaki:
<p>Na začetku ponovimo in osvežimo snov prejšnje ure.</p> <p>Dijakom predstavimo pojem razhroščevanja (debugging). Kot zanimivost povemo anekdoto o nastanku pojma hrošč v računalništvu.</p> <p>Predstavimo tehnike razhroščevanja:</p> <ul style="list-style-type: none"><li>• zapisovanje vrednosti, ki se spreminjajo,</li><li>• izpis vrednosti na standardni izhod,</li><li>• analiza podatkov po delovanju programa,</li><li>• razvojna okolja z vgrajenimi sistemi za razhroščevanje.</li></ul> <p>Pripravimo primer z diagramom poteka – naredimo primer diagrama poteka, v katerem namerno naredimo eno ali več napak. Dijake spodbudimo, da najdejo napako z uporabo tehnik razhroščevanja.</p> <p>Skupaj ugotovimo, da so tehnike razhroščevanja popolnoma vsakdanje tehnike odkrivanja napak; primer: matematično nalogo smo rešili narobe; postopek iskanja napake.</p> <p>V spletni učilnici imamo pripravljen program v programski opremi EV3, ki vsebuje napake. Dijake spodbudimo, da si program prenesejo in s tehnikami razhroščevanja poskusijo odkriti ter odpraviti napake, da bo program deloval pravilno.</p> <p>Naslednjo uro sledijo praktične vaje.</p>	<p>Poslušajo, sodelujejo, zapisujejo.</p> <p>Poslušajo, sodelujejo, zapisujejo.</p>

**UČNA PRIPRAVA**Predmet: **IP Informatika**

Program: gimnazija

Enota: **LEGO Mindstorms**Tema: **Razhroščevanje (debugging)**

Tip učne enote: kombinirana

Zap. št.  
priprave:

15

Oddelek:

Datum:

Zap. št. ure:

29–30

Šola:

Učitelj:

Šolsko leto:

Operativni vzgojno-izobraževalni cilji:

Dijaki utrdijo do sedaj pridobljeno znanje in tehnike razhroščevanja.

Učne oblike: kombinirana

Učne metode: razlaga, diskusija, reševanje problemov

Viri:

navodila za uporabnike LEGO Mindstorms EV3  
[https://le-www-live-s.legocdn.com/sc/media/files/user-guides/ev3/ev3\\_user\\_guide\\_us-ab5d4fc71211c1edc77a7362bab2d88a.pdf?la=en-us](https://le-www-live-s.legocdn.com/sc/media/files/user-guides/ev3/ev3_user_guide_us-ab5d4fc71211c1edc77a7362bab2d88a.pdf?la=en-us)

Učni

pripomočki: tabla, flomastri različnih barv, osebni računalnik, osnovni komplet LEGO Mindstorms EV3

Didaktična struktura ure:

1. Uvod (5 min) – uvodna motivacija
2. Ponovitev razhroščevanja (10 min)
3. Reševanje problemov (60 min)
4. Diskusija o rešitvah (15 min)

Učitelj:	Dijaki:
Na začetku ponovimo in osvežimo snov prejšnje ure.	Poslušajo, sodelujejo, zapisujejo.
Z dijaki osvežimo pojem razhroščevanja (debugging).	
Za dijake pripravimo več različno zahtevnih izzivov in jih spodbudimo k njihovem reševanju.	Programirajo, razmišljajo, rešujejo probleme.
Dijake usmerjamo do pravih rešitev in jih spodbujamo k ustvarjalnosti ter redoljubnemu sestavljanju programa. Ob napakah jih spodbujamo k razhroščevanju.	
Naslednjo uro sledijo funkcije (metode).	

**UČNA PRIPRAVA**Predmet: **IP Informatika**

Program: gimnazija

Enota: **LEGO Mindstorms**Tema: **Funkcije (metode)**

Tip učne enote: kombinirana

Zap. št.  
priprave:

16

Oddelek:

Datum:

Zap. št. ure:

31–32

Šola:

Učitelj:

Šolsko leto:

Operativni vzgojno-izobraževalni cilji:

Dijaki osvojijo pojem funkcije (metode).

Dijaki osvojijo gradnjo svojih funkcij v programski opremi EV3.

Učne oblike: kombinirana

Učne metode: razlaga, diskusija, reševanje problemov

Viri:

navodila za uporabnike LEGO Mindstorms EV3

[https://le-www-live-s.legocdn.com/sc/media/files/user-guides/ev3/ev3\\_user\\_guide\\_us-ab5d4fc71211c1edc77a7362bab2d88a.pdf?la=en-us](https://le-www-live-s.legocdn.com/sc/media/files/user-guides/ev3/ev3_user_guide_us-ab5d4fc71211c1edc77a7362bab2d88a.pdf?la=en-us)

Učni

pripomočki: tabla, flomastri različnih barv, osebni računalnik, osnovni komplet LEGO Mindstorms EV3

Didaktična struktura ure:

1. Uvod (5 min) – uvodna motivacija
2. Predstavitev funkcij (15 min)
3. Primeri funkcij, ki jih že poznajo (15 min)
4. Gradnja svoje funkcije v EV3 (30 min)
5. Diskusija in priprava na preverjanje znanja (25 min)



Učitelj:	Dijaki:
<p>Na začetku ponovimo in osvežimo snov prejšnje ure.</p> <p>Dijakom pojasnimo pojem funkcije (metode) v programiranju ter pojem vhodnih (argumentov) in izhodnih podatkov (rezultat).</p> <p>Dijakom predstavimo primere funkcij, ki jih že poznajo:</p> <ul style="list-style-type: none"> <li>• seštevanje,</li> <li>• množenje,</li> <li>• sinus, kosinus,</li> <li>• dolžina seznama,</li> <li>• itd.</li> </ul> <p>Z dijaki ugotovimo, kateri so vstopni in izstopni podatki.</p> <p>Dijake spodbudimo k razmišljanju, kdaj so funkcije uporabne in zakaj.</p> <p>Dijakom predstavimo tehniko grajenja funkcije v programski opremi EV3 (My Block).</p> <div data-bbox="245 999 1165 1691" data-label="Diagram"> </div> <p>Dijake spodbudimo k diskusiji o preverjanju znanja, ki sledi naslednjo uro.</p>	<p>Poslušajo, sodelujejo, zapisujejo.</p>

**UČNA PRIPRAVA**Predmet: **IP Informatika**

Program: gimnazija

Enota: **LEGO Mindstorms**Tema: **Teoretično preverjanje znanja**

Tip učne enote: kombinirana

Zap. št.  
priprave:

17

Oddelek:

Datum:

Zap. št. ure:

33–34

Šola:

Učitelj:

Šolsko leto:

Operativni vzgojno-izobraževalni cilji:

Dijaki pokažejo do sedaj osvojeno teoretično znanje.

Učne oblike: kombinirana

Učne metode: razlaga, diskusija, reševanje problemov

Viri:

Učni

pripomočki: osebni računalnik, spletna učilnica

Didaktična struktura ure:

1. Reševanje preverjanja teoretičnega znanja (45 min)
2. Diskusija o rešitvah preverjanja znanja (45 min)

Učitelj:	Dijaki:
<p>Dijakom smo pripravili vprašanja in naloge, s katerimi bomo preverili njihovo teoretično znanje, ki so ga do sedaj pridobili pri predmetu Informatika.</p> <p>Vprašanja zberemo v spletni učilnici, kjer dijaki tudi rešujejo preverjanje. Naloge naj bodo različne:</p> <ul style="list-style-type: none"> <li>• zaprti tip vprašanj,</li> <li>• povezovanje,</li> <li>• zapolnjevanje manjkajočih členov,</li> <li>• računske naloge,</li> <li>• itd.</li> </ul> <p>Naloge naj bodo različno vrednotene glede na težavnost; vsak dijak naj dobi enako število različno vrednotenih nalog.</p> <p>Pripravimo več nalog in jih razdelimo v sklope, nato pa nastavimo naključno izbiranje nalog iz vsakega sklopa.</p> <p>Vprašanja naj zavzemajo sledeče teme:</p> <ul style="list-style-type: none"> <li>• osnovni pojmi,</li> <li>• algoritmi,</li> <li>• diagrami poteka,</li> <li>• lastnosti motorjev in senzorjev,</li> <li>• lastnosti blokov,</li> <li>• osnovni principi programiranja EV3,</li> <li>• razhroščevanje,</li> <li>• itd.</li> </ul> <p>Med naloge vključimo tudi primere programov (tako diagramov poteka kot EV3). Dijaki naj ugotavljajo, kaj je shranjeno v spremenljivko v določenem trenutku in kaj vrača program. Ena izmed nalog naj vsebuje program z napako, ki naj jo dijaki najdejo in odpravijo.</p> <p>V drugi uri z dijaki predebatiramo pravilne rešitve vprašanj in nalog iz preverjanja znanja ter njihove rešitve.</p> <p>Naslednjo uro sledi komuniciranje med roboti.</p>	<p>Rešujejo naloge na računalniku.</p>

**UČNA PRIPRAVA**Predmet: **IP Informatika**

Program: gimnazija

Enota: **LEGO Mindstorms**Tema: **Komuniciranje**

Tip učne enote: kombinirana

Zap. št.  
priprave:

18

Oddelek:

Datum:

Zap. št. ure:

35–36

Šola:

Učitelj:

Šolsko leto:

Operativni vzgojno-izobraževalni cilji:

Dijaki osvojijo tehnike komuniciranja med roboti.

Učne oblike: kombinirana

Učne metode: razlaga, diskusija, reševanje problemov

Viri:


navodila za uporabnike LEGO Mindstorms EV3  
[https://le-www-live-s.legocdn.com/sc/media/files/user-guides/ev3/ev3\\_user\\_guide\\_us-ab5d4fc71211c1edc77a7362bab2d88a.pdf?la=en-us](https://le-www-live-s.legocdn.com/sc/media/files/user-guides/ev3/ev3_user_guide_us-ab5d4fc71211c1edc77a7362bab2d88a.pdf?la=en-us)

Učni

pripomočki: tabla, flomastri različnih barv, osebni računalnik, osnovni komplet LEGO Mindstorms EV3

Didaktična struktura ure:

1. Uvod (5 min) – uvodna motivacija
2. Predstavitev možnosti komunikacije med roboti (20 min)
3. Reševanje problemov (60 min)
4. Diskusija (5 min)

Učitelj:	Dijaki:
<p>Dijakom predstavimo možnosti komuniciranja med roboti.</p> <p>Dijakom predstavimo bloke, ki so namenjeni komunikaciji robotov:</p> <ul style="list-style-type: none"><li>• Bluetooth connection,</li><li>• messaging.</li></ul>  <p>Dijake spodbudimo, da se v skupinah povežejo med sabo in skupaj napišejo programe, ki bodo komunicirali med sabo.</p> <p>Dijakom pripravimo nekaj nalog, ki vključujejo povezovanje in sodelovanje robotov.</p> <p>Naslednjo uro sledi zadnji izziv.</p>	<p>Poslušajo, sodelujejo, zapisujejo.</p> <p>Programirajo, sodelujejo, raziskujejo.</p>

**UČNA PRIPRAVA**Predmet: **IP Informatika**

Program: gimnazija

Enota: **LEGO Mindstorms**Tema: **Izziv**

Tip učne enote: kombinirana

Zap. št.  
priprave:

19

Oddelek:

Datum:

Zap. št. ure:

37–38

Šola:

Učitelj:

Šolsko leto:

Operativni vzgojno-izobraževalni cilji:

Dijaki rešujejo izziv, ki vključuje vsa znanja, ki so jih pridobili do sedaj.

Učne oblike: kombinirana

Učne metode: razlaga, diskusija, reševanje problemov

Viri:

navodila za uporabnike LEGO Mindstorms EV3  
[https://le-www-live-s.legocdn.com/sc/media/files/user-guides/ev3/ev3\\_user\\_guide\\_us-ab5d4fc71211c1edc77a7362bab2d88a.pdf?la=en-us](https://le-www-live-s.legocdn.com/sc/media/files/user-guides/ev3/ev3_user_guide_us-ab5d4fc71211c1edc77a7362bab2d88a.pdf?la=en-us)

Učni

pripomočki: tabla, flomastri različnih barv, osebni računalnik, osnovni komplet LEGO Mindstorms EV3

Didaktična struktura ure:

1. Uvod (5 min) – uvodna motivacija
2. Predstavitev izziva (10 min)
3. Reševanje izziva (70 min)
4. Diskusija (5 min)

Učitelj:	Dijaki:
<p>Dijakom predstavimo izziv.</p> <p>Primer izziva:</p> <p><b>Naloga:</b> Naredili boste teptalnik (ratrak). Ratrak za stik s podlago običajno uporablja gosenice, kar lahko na svojem robotu prikažete tudi vi. Prikazali boste osnovno delovanje ratraka na smučišču. Ratrak se torej vozi po smučišču in tepta oziroma valja snežno podlago ter jo tako pripravlja za smučarje. To dela z vožnjo gor in dol po smučišču. Vaša naloga je, da sprogramirate robota, da se bo vozil sem ter tja in sproti opravljal še nekaj drugih nalog.</p> <p><b>Vožnja:</b> Robot naj se torej pelje naravnost. Ko pride do ovire, naj sekundo počaka na vaš znak. Robotu lahko podate znak s tipkama levo/desno na robotu. Če robot ne dobi znaka, naj gre desno. Nato se premakne za svojo širino in zopet zavije v smer, ki je bila izbrana v prejšnjem koraku (ali ste jo določili s tipkami ali pa jo je izbral sam). Robot zopet pelje do ovire in ponovi korak.</p> <p><b>Zvok:</b> Bolj kot je strm klanec, glasnejši je motor ratraka. Tudi naš robot deluje na podoben način. Glede na strmost poti naj ob vožnji spušča zvoke ustrezne frekvence/tonske višine/glasnosti. Ko se začne približevati oviri (npr. 20 cm prej), naj namesto zvoka motorja oponaša zvok parkirnih senzorjev – višina zvoka naj se stopnjuje z bližino ovire. Ob obratih uporabite zvoke, ki ste jih uporabljali že pri prejšnjih izzivih.</p> <p><b>Slika:</b> Robota dopolnite tudi vizualno. Ker se bliža novo leto, naj ves čas na robotu utripajo lučke v izmenjujočih se barvah. Ob akcijah (ovira, obrat ...) robota opremite tudi z izrazi na zaslonu.</p> <p><b>Prekinitev:</b> Robot naj ima priključen tudi touch senzor. Če v katerem koli trenutku pritisnete touch senzor, naj se robot ustavi za 2 sekundi. Če v teh dveh sekundah ponovno pritisnete gumb, naj se program prekine (ratrak se ugasne). Na smučišču imamo tudi prepade, na kar morate biti še posebej pozorni. Če robot zapelje čez rob mize (oziroma prepada na smučišču), naj se ustavi in obrne.</p> <p><b>Namig:</b> Strmino določite z žiro senzorjem, dobljeno vrednost pa povežite z zvočnikom. Oddaljenost izmerite z ultrazvočnim senzorjem, dobljeno vrednost pa povežite z zvočnikom. Light senzor meri tudi količino odbite svetlobe. Če pod senzorjem ni nobene površine, je količina odbite svetlobe enaka 0.</p>	<p>Poslušajo, sodelujejo, zapisujejo.</p> <p>Razmišljajo, sodelujejo, rešujejo izziv.</p>

Učitelj:	Dijaki:
Dijake usmerjamo do pravilne rešitve in jih spodbujamo k ustvarjalnosti ter redoljubnemu sestavljanju programa.	